

Oracle® Cloud

Using Oracle Infrastructure Monitoring



E73189-54
September 2022



Oracle Cloud Using Oracle Infrastructure Monitoring,
E73189-54

Copyright © 2016, 2022, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	x
Documentation Accessibility	x
Related Resources	x
Conventions	xi

1 Getting Started with Oracle Infrastructure Monitoring

About Oracle Infrastructure Monitoring	1-1
About Oracle Infrastructure Monitoring Roles and Users	1-2
Configure Oracle Infrastructure Monitoring	1-3

2 Add Entities for Infrastructure Monitoring

Supported Entity Types	2-1
Prerequisites and Monitoring Credentials	2-5
Enable Host Monitoring	2-7
Add Entities from the Console	2-9
Define Associations Between Entities	2-12
Discover Cloud Services	2-13
Add Entities Using JSON Files	2-23
Download and Customize Oracle Infrastructure Monitoring JSONs	2-29
Use Tags to Define Associations Between Entities	2-32

3 Enable Monitoring for Previously Discovered Entities

Auto-Discovery of Previously Discovered Entities	3-1
Manual Discovery of Log Analytics Entities	3-4

4 Extend Monitoring Capability with Metric Collectors

Expand Monitoring with collectd	4-1
Example Read Plugin — Processes Plugin	4-2

Example: Configuring collectd	4-2
Example: Generic Metric Collector Entity Type (Auto-mapping)	4-5
Expand Monitoring with Telegraf	4-7
Example Input Plugin: Processes Plugin	4-8
Example: Configure Telegraf for Oracle Management Cloud Integration	4-8
Example Generic Metric Collector Entity (Telegraf)	4-10
Troubleshooting Telegraf Metric Collection	4-15

5 Expand Monitoring Capability with Custom Metrics

Custom Metric Lifecycle	5-2
Working with Custom Metrics	5-4
Creating Custom Metrics for MySQL and SQL Server Databases	5-10

6 Host Process Monitoring

Create a Process Set	6-2
Map the Process Set ID to One or More Hosts	6-3
Monitor the Processes from the UI	6-4
Create Alert Rules to Monitor Process Status and Resource Consumption	6-6

7 Set Up Alert Rules

Typical Workflow for Setting Up Alert Rules	7-1
Set Up Alert Thresholds and Notifications	7-1
Create an Alert Rule	7-2
Set Up Notification Channels	7-6
Create an Email Notification Channel	7-6
Create a Mobile Notification Channel	7-7
Create a WebHook Notification Channel (Integration)	7-8
Create a PagerDuty Notification Channel	7-10
Create a ServiceNow Notification Channel	7-13
Create a Slack Notification Channel	7-16

8 Monitor the Availability and Performance of Your Infrastructure

Typical Workflow for Monitoring the Availability and Performance of Your Infrastructure	8-1
Monitor Availability Status	8-1
Investigate Alerts	8-3
Monitor Availability Status Within Each Tier	8-6
Monitor Performance Within Each Tier	8-7
Monitor Entity Health	8-9

9 Oracle Infrastructure Monitoring Administration Tasks

Typical Administration Tasks for Oracle Infrastructure Monitoring	9-1
Maintenance Windows	9-1
Change Monitoring Configuration	9-2
Create and Set Global Properties	9-2
Creating a Global Property	9-3
Setting the Value of a Global Property	9-3
Delete Entities	9-3
Delete Entities from the Administration Console	9-3
View Deleted Entities	9-4
Delete Entities Using omcli	9-4

10 Troubleshooting

Lack of Data	10-1
Create an Agent Support Bundle	10-5
Host Prerequisite Validation	10-6
Status Unknown	10-11
Database Status is Shown as Down when the Database is Up	10-11

A Monitoring Prerequisites and Credentials

Host	A-1
Docker Engine / Docker Container	A-2
XEN Virtual Platform / XEN Virtual Server	A-4
Oracle Database	A-5
AWS-RDS Oracle DB	A-10
Oracle Automatic Storage Management (ASM)	A-10
Oracle NoSQL	A-11
MySQL Database	A-11
Microsoft SQL Server	A-12
MongoDB Database	A-13
Oracle WebLogic Server (includes WebLogic Domain and WebLogic Cluster)	A-14
Oracle Service Bus	A-15
Tomcat	A-16
Oracle Traffic Director (OTD)	A-17
Apache HTTP Server	A-18
Oracle HTTP Server (OHS)	A-19
Arista Ethernet Switch	A-19

Cisco Ethernet (Catalyst) Switch	A-20
Cisco Nexus Ethernet Switch	A-20
Oracle Power Distribution Unit (PDU)	A-20
Juniper Ethernet Switch	A-20
Oracle Infiniband Switch	A-21
Brocade Fibre Channel Switch	A-21
SCOM (System Center Operations Manager)	A-21
Juniper SRX Firewall	A-22
Fujitsu Server	A-22
Intel/SPARC Computers	A-22
VMware vCenter	A-23
Docker Swarm	A-24
Apache SOLR	A-25
Hadoop Cluster	A-25
Arbor TMS/CP	A-26
Juniper Netscreen Firewall	A-26
Juniper MX Router	A-26
F5 BIG-IP LTM	A-27
F5 BIG-IP DNS	A-27
ES2 Ethernet Switch	A-27
Oracle Flash Storage	A-27
Apache Cassandra DB	A-28
Oracle VM Server for SPARC (LDoms)	A-29
Coherence	A-29
Oracle Unified Directory(ODU)	A-30
Oracle Access Manager (OAM)	A-31
Oracle Internet Directory (OID)	A-31
Microsoft Internet Information Services (IIS)	A-32
Oracle Identity Manager (OIM)	A-34
Oracle Clusterware (CRS)	A-34
JBOSS	A-34
Kubernetes Cluster	A-36
Oracle GoldenGate	A-49
Oracle VM Manager	A-50
Oracle JVM Runtime	A-50
Microsoft Azure	A-51
Apache Kafka	A-53

B Entity Attributes and Properties

C Discovery

Add Apache HTTP Server	C-3
Add Apache SOLR	C-6
Add Apache Zookeeper	C-9
Add Arbor CP	C-10
Add Arbor TMS	C-13
Add Arista Ethernet Switch	C-15
Add Brocade Fibre Channel Switch	C-17
Add Apache Cassandra Database	C-21
Add Cisco Catalyst Switch	C-23
Add Cisco Nexus Ethernet Switch	C-26
Add Docker Engine/Docker Container	C-30
Add Docker Swarm	C-34
Add F5 BIG-IP DNS	C-37
Add F5 BIG-IP LTM	C-39
Add Hadoop Cluster	C-41
Add JBoss Server/Domain	C-44
Add Juniper Ethernet Switch	C-49
Add Juniper MX Router	C-52
Add Juniper Netscreen Firewall	C-54
Add Juniper SRX Firewall	C-56
Add Kubernetes Cluster	C-59
Add Microsoft IIS	C-62
Add Microsoft SCOM	C-66
Add Microsoft SQL Server	C-69
Add MongoDB	C-73
Add MySQL Database	C-75
Add NetApp FAS	C-78
Add NGINX	C-79
Add Oracle Access Manager/Oracle Internet Directory	C-81
Add Oracle Automatic Storage Management (ASM)	C-82
Add Oracle Clusterware (CRS)	C-83
Add Oracle Coherence Clusters	C-85
Add Oracle Database Listener Cluster	C-87
Add Oracle Database Listeners	C-88
Add Oracle Databases	C-90
Add Oracle Database Systems	C-98
Add Oracle ES2 Ethernet Switches	C-107
Add Oracle GoldenGate	C-110
Add Oracle HTTP Server	C-115

Add Oracle Identity Manager	C-117
Add Oracle Infiniband Switch	C-119
Add Oracle JVM Runtime	C-122
Add Oracle NoSQL Database	C-124
Add Oracle Pluggable Database	C-126
Add Oracle Power Distribution Unit (PDU)	C-127
Add Oracle Service Bus	C-129
Add Oracle Traffic Director	C-131
Add Oracle Unified Directory	C-135
Add Oracle Virtual Networking	C-137
Add Oracle VM Manager	C-139
Add Oracle VM Server for SPARC (LDOMS)	C-141
Add Oracle WebLogic Server/Domain	C-143
Add SPARC/Intel Computers	C-145
Add Tomcat	C-147
Add VMware vCenter	C-152
Add ZFS Storage Appliance	C-153

D Agent-monitored Entity Types and Cloud Services

E Monitor AWS - RDS Oracle DB

F Configure a Coherence Cluster

G Additional collectd Configurations and Information

Manual Mapping	G-1
Example: Generic Metric Collector Entity (Manual Mapping)	G-1
Example: Mapping Metadata	G-4
Example: Destination Metric Definitions	G-5
Metric Schema Mapping (collectd)	G-8
Send a Subset of collectd Metrics to Oracle Management Cloud	G-10
Receive Metrics from a Remote Generic Metric Collector	G-11
Availability (Up/Down) Status for Entities Monitored by collectd	G-14
Troubleshooting collectd Metric Collection	G-15

H Additional Telegraf Configurations and Information

Metric Schema Mapping (Telegraf)	H-1
----------------------------------	-----

Receive Metrics from a Remote Telegraf Collector	H-4
Availability (Up/Down) Status for Entities Monitored by Telegraf	H-6

| Custom Metric Collection Methods and Metric Columns

OS Command	I-1
SQL Query	I-2
Java Management Extensions (JMX)	I-3
REST	I-4
Compute Expressions	I-6
Rate and Delta Metric Columns	I-9

Preface

Oracle Infrastructure Monitoring provides performance and availability monitoring for your enterprise.

Topics:

- [Audience](#)
- [Related Resources](#)
- [Conventions](#)

Using Oracle Infrastructure Monitoring describes how to use this service to perform common monitoring and alerting tasks.

Audience

Using Oracle Infrastructure Monitoring is intended for administrators who want to set up status and performance infrastructure monitoring as well as alerting across their enterprise.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Resources

For more information about Oracle Management Cloud see:

- [Oracle Cloud](#)

Conventions

Table 1 Text Conventions

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates the book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph. URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with Oracle Infrastructure Monitoring

Topics:

- [About Oracle Infrastructure Monitoring](#)
- [About Oracle Infrastructure Monitoring Roles and Users](#)
- [Configure Oracle Infrastructure Monitoring](#)

About Oracle Infrastructure Monitoring

Oracle Infrastructure Monitoring is built on Oracle Management Cloud's secure, unified data platform and provides proactive monitoring for your entire IT infrastructure. As an administrator, you can perform status and health monitoring across tiers and be alerted about issues, troubleshoot and resolve them before they affect users.

Oracle Infrastructure Monitoring simplifies monitoring by offering a common set of metrics that allow you to compare performance across various vendor technologies. The functionality also extends to vendor-specific metrics to monitor unique vendor functionality. In addition, this service automatically generates alerts when managed entities are down and allows you to create alert rules that specify the metrics thresholds and notifications options.

Here are some common terms and concepts used in Oracle Management Cloud and Oracle Infrastructure Monitoring.

Entity: A monitored resource, such as, a database, a host server, a compute resource, or an application server.

Metrics: A set of parameters and values measured and collected periodically for a particular system for tracking performance and availability. For a complete list and description of all metrics collected for each entity, see List of Supported Entities in the *Metric Reference for Oracle Infrastructure Monitoring*.

Thresholds: Boundary values that the monitored metric values are compared against. If a metric value crosses a set threshold, then, an alert is raised.

Alerts: Information generated in response to an availability issue or when a metric crosses its thresholds. Conditions for generating alerts are defined in Alert Rules. Alerts sent to administrators by using various channels, such as, email and SMS are known as **notifications**.

Alert rules: A set of rules that define the conditions under which alerts are generated and notifications sent when an alert is raised. Alert conditions consist of a metric, a comparison operator, and thresholds against which metric values are evaluated.

Cloud Agent: The on-premises interface to Oracle Management Cloud that is configured to monitor various entities by collecting status, performance and configuration data.

Gateway: A gateway is a Cloud Agent that acts as a proxy between Oracle Management Cloud and all other Cloud Agents.

Using the Oracle Infrastructure Monitoring, you can:

- Monitor your entire IT infrastructure from a single platform
- Monitor availability and performance across a broad range of infrastructure technologies
- Identify potential performance issues within a tier
- Set up alert rules to notify you of availability and performance issues

About Oracle Infrastructure Monitoring Roles and Users

Once you are an Oracle Cloud customer and you create an Oracle Management Cloud instance, the following user roles are provisioned:

- Oracle Management Cloud Administrator
- Oracle Management Cloud User

Table 1-1 Roles for Oracle Infrastructure Monitoring

Role	Tasks
Oracle Management Cloud Administrator	<ul style="list-style-type: none"> • Set up infrastructure monitoring by deploying and configuring the gateway and cloud agents. • Manage Cloud Agents. • Add entities to be monitored. • Configure alert rules. • Delete entities. • Disable notifications on alerts (during maintenance periods). • View and monitor infrastructure status and performance. • Receive alert notifications and view alerts.
Oracle Management Cloud User	<ul style="list-style-type: none"> • View and monitor infrastructure status and performance. • Receive alert notifications and view alerts.

For more information about the tasks that the users assigned with the above roles can perform, see *Add Users and Assign Roles* in *Getting Started with Oracle Management Cloud*.

Note:

If you are using an older version of Oracle Management Cloud (prior to V4), you can enable or disable individual services on entities. See *Previously Provisioned Tenants: Enabling or Disabling Services on Entities*. This capability does not apply to newer versions of Oracle Management Cloud. See *Enabling License Editions*.

Configure Oracle Infrastructure Monitoring

Oracle Infrastructure Monitoring uses Cloud Agents to monitor entities for availability status and performance. Cloud Agents are made aware of entities they need to monitor through the process of *adding entities*. As an Oracle Infrastructure Monitoring Administrator, perform the following tasks to add entities to your monitoring service:

Table 1-2 Typical Workflow for Adding Oracle Infrastructure Monitoring Entities

Task	Description	More Information
Pre-requisite: Deploy Cloud Agents	Cloud Agents are deployed for multiple Oracle Management Services that require an agent to enable specific functionality. The agents deployment task is part of the initial set up of your service. It includes the deployment of Cloud Agents as well as an optional gateway that acts as a proxy between Oracle Management Cloud and all Cloud Agents.	See Installing Oracle Management Cloud Agents in <i>Installing and Managing Oracle Management Cloud Agents</i> .
Decide what you want to monitor.	Oracle Management Cloud lets you monitor a wide variety of entity types across your IT environment. Identify the entity types you intend to monitor	Agent-monitored Entity Types and Cloud Services
Prepare your entities for monitoring.	Most entities require some configuration or specific credentials in order to enable their monitoring. Once you've identified the types of entities to monitor, perform the steps required to allow monitoring on those entities.	Prerequisites and Monitoring Credentials
Decide how you want to add entities to Infrastructure Monitoring.	Entities can be added to Infrastructure Monitoring in two ways: <ul style="list-style-type: none">• Directly from the Oracle Management Cloud UI.• Using JSON files.	.

Table 1-2 (Cont.) Typical Workflow for Adding Oracle Infrastructure Monitoring Entities


Task	Description	More Information
<p><i>IF you add entities from the UI:</i></p>	<p>The Oracle Management Cloud console provides an intuitive interface that simplifies adding one or a small number of entities.</p> <div data-bbox="776 489 971 1234" style="border-left: 1px solid #0070C0; border-right: 1px solid #0070C0; border-bottom: 1px solid #0070C0; padding: 10px; margin-top: 20px;"> <p> Note:</p> <p>Although not all entity types can be added using the UI, the list of UI-enabled entity types that can be added increases with each release.</p> </div>	<p>Add Entities from the Console</p>

Table 1-2 (Cont.) Typical Workflow for Adding Oracle Infrastructure Monitoring Entities

Task	Description	More Information
<p>IF <i>you add entities using JSON files:</i></p>	<p>Using JSON files. JSON files are used by the Cloud Agent to discover and monitor entities. Adding entities via JSON files lets automate the process of adding entities</p> <ol style="list-style-type: none"> 1. Download and edit the sample entity definition and credentials JSON files. Oracle provides sample JSON files to help in defining your entities. Download and edit these sample files with the information about your entities and their credentials. 2. Add entities to be monitored by the Oracle Infrastructure Monitoring and verify their state. Use the <code>omcli</code> command line interface and your customized JSON files to add your entities to the monitoring service and verify their status. 	<p>Download and Customize Oracle Infrastructure Monitoring JSONs Add Entities Using JSON Files</p>

2

Add Entities for Infrastructure Monitoring

Adding entities to Oracle Infrastructure Monitoring lets you monitor their performance and availability via Cloud agents.

Topics:

- [Supported Entity Types](#)
- [Prerequisites and Monitoring Credentials](#)
- [Enable Host Monitoring](#)
- [Add Entities from the Console](#)
- [Discover Cloud Services](#)
- [Add Oracle Database Systems](#)
- [Add Entities Using JSON Files](#)

Supported Entity Types

Oracle Infrastructure Monitoring lets you monitor a wide range of entity types. The first step in setting up Oracle Infrastructure Monitoring is to determine whether the types of entities you want to monitor are currently supported. Some entity types require additional configuration before they can be monitored. See [Prerequisites and Monitoring Credentials](#) for more information.



Note:

For the latest information on supported entities and other important release information, see [Infrastructure Monitoring Cloud Service Master Note \(Doc ID 2195015.1\)](#).

The following table lists all entity types currently supported by Oracle Infrastructure Monitoring.

For a complete list of entity type properties (JSON and UI), see [Entity Attributes and Properties](#).

Table 2-1 Supported Entity Types

Entity Type	Additional Information
Apache Hadoop	Add Hadoop Cluster
Apache Kafka	Apache Kafka can only be added as part of Zookeeper discovery and not by itself. See Add Apache Zookeeper for instructions.
Apache HTTP Server	Add Apache HTTP Server

Table 2-1 (Cont.) Supported Entity Types

Entity Type	Additional Information
Apache SOLR	Add Apache SOLR
Apache Zookeeper	Add Apache Zookeeper
Arbor CP	Add Arbor CP
Arbor TMS	Add Arbor TMS
Arista Ethernet Switch	Add Arista Ethernet Switch
Brocade Fibre Channel Switch	Add Brocade Fibre Channel Switch
Cassandra Database	Add Apache Cassandra Database
Cisco Catalyst Switch	Cisco Ethernet (Catalyst) Switch
Cisco Nexus Ethernet Switch	Add Cisco Nexus Ethernet Switch
Docker Engine, Docker Container	Add Docker Engine/Docker Container
Docker Swarm	Add Docker Swarm
F5 BIG-IP LTM	Add F5 BIG-IP LTM
F5 BIG-IP GTM	Add F5 BIG-IP DNS
Hosts <ul style="list-style-type: none"> • Linux • Solaris • AIX • Windows 	The operating system user used to install the Cloud Agent is also used as the host monitoring credential. Your hosts are automatically added as entities when a Cloud Agent is installed. However, hosts are not automatically monitored. To enable monitoring for host entities, see Enable Host Monitoring .
Juniper Ethernet Switch	Add Juniper Ethernet Switch
Juniper MX Router	Add Juniper MX Router
Juniper Netscreen Firewall	Add Juniper Netscreen Firewall
Juniper SRX Firewall	Add Juniper SRX Firewall
JBoss	Add JBoss Server/Domain
Kubernetes	Add Kubernetes Cluster
Microsoft SCOM Integration: Windows (Host) servers, SQL Server, Hyper V, Active Directory, Microsoft Exchange	Add Microsoft SCOM
Microsoft IIS	Add Microsoft IIS (Remote monitoring using remote WMI.)
Microsoft SQL Server	Add Microsoft SQL Server
MongoDB	Add MongoDB Most MongoDB metrics can be remotely collected, except for metrics which require a local agent to be deployed on the same host as MongoDB. These metrics include: CPU Usage, CPU Utilization, Memory Usage, Memory Utilization, Total CPUs, Total Memory.

Table 2-1 (Cont.) Supported Entity Types





Entity Type	Additional Information
MySQL Database	Add MySQL Database Most MySQL metrics can be remotely collected, except for metrics which require a local agent to be deployed on the same host as MySQL. These metrics include: CPU Usage, CPU Utilization, Memory Usage, Memory Utilization, Total CPUs, Total Memory.
NetApp FAS	Add NetApp FAS
NGINX	Add NGINX
Oracle Access Manager	Add Oracle Access Manager/Oracle Internet Directory
Oracle Automatic Storage Management (ASM)	Add Oracle Automatic Storage Management (ASM)
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Consider adding this entity as part of a Database System. See Add Oracle Database Systems.</p> </div>
Oracle Clusterware	Add Oracle Clusterware (CRS)
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Consider adding this entity as part of a Database System. See Add Oracle Database Systems.</p> </div>
Oracle Coherence	Add Oracle Coherence Clusters
Oracle Database System and Oracle Database	Adding a database system lets you add all components that make up a logical database group (a database and listener, or perhaps a database, listener, and ASM) as a single entity instead of adding the components individually. See Add Oracle Database Systems for more information. Alternatively, you can still add Oracle database entities individually. See Add Oracle Databases .
Oracle Database Listener	Add Oracle Database Listeners Remote monitoring for the Oracle database listener requires SSH credentials and is supported on Linux, AIX, Solaris.
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Consider adding this entity as part of a Database System. See Add Oracle Database Systems.</p> </div>

Table 2-1 (Cont.) Supported Entity Types

Entity Type	Additional Information
Oracle Database Cluster Listener	Add Oracle Database Listener Cluster Remote monitoring for Linux.
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Consider adding this entity as part of a Database System. See Add Oracle Database Systems.</p> </div>
Oracle Ethernet Switch ES2	Add Oracle ES2 Ethernet Switches
Oracle GoldenGate	Add Oracle GoldenGate
Oracle HTTP Server (Standalone)	Add Oracle HTTP Server
Oracle HTTP Server (Managed)	Add Oracle HTTP Server
Oracle Identity Manager	Add Oracle Identity Manager
Oracle InfiniBand Switch	Add Oracle Infiniband Switch
Oracle Internet Directory	Add Oracle Access Manager/Oracle Internet Directory
Oracle Java Virtual Machine (JVM)	Add Oracle JVM Runtime
Oracle NoSQL	Add Oracle NoSQL Database While performance metrics can be remotely collected, some configuration metrics required a local agent to be installed in one of the NoSQL nodes.
Oracle PeopleSoft	Oracle Peoplesoft: Set Up the Environment
Oracle Pluggable Database (PDB)	Add Oracle Pluggable Database To monitor a PDB, it must be in <i>non-restricted</i> mode. Alternatively, an administrative user can be granted <i>restricted session</i> privileges, however, this is not optimal.
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Consider adding this entity as part of a Database System. See Add Oracle Database Systems.</p> </div>
Oracle Power Distribution Unit (PDU)	Add Oracle Power Distribution Unit (PDU)
Oracle Service Bus	Add Oracle Service Bus

Table 2-1 (Cont.) Supported Entity Types

Entity Type	Additional Information
Oracle SOA Infrastructure	See Add Oracle WebLogic Server/Domain Oracle SOA Infrastructure entities are automatically discovered as part of the WebLogic Domain discovery.
<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <p>When you add a WebLogic Domain entity (requiring credentials), because Oracle Management Cloud connects to the WebLogic Admin Server, all WebLogic Clusters and WebLogic Servers that are part of that domain are automatically discovered. There's no need to add them separately.</p> </div>	
Oracle Traffic Director (OTD), Oracle Traffic Director Configuration	Add Oracle Traffic Director For OTD Configuration, the cloud agent needs to be installed on the host where the OTD Admin Server resides, and the OTD Instances can be monitored remotely.
Oracle Unified Directory	Add Oracle Unified Directory
Oracle VM Manager	Add Oracle VM Manager
Oracle Virtual Networking	Add Oracle Virtual Networking
Oracle VM Server for SPARC (LDoms)	Add Oracle VM Server for SPARC (LDOMS)
Oracle WebLogic Server (includes WebLogic Domain, WebLogic Cluster, WebLogic Node Manager)	Add Oracle WebLogic Server/Domain
Siebel	Siebel: Set up the Environment
SPARC/Intel Computers (h/w)	Add SPARC/Intel Computers
Tomcat	Add Tomcat
VMware vCenter integration	Add VMware vCenter
ZFS Storage Appliance	Add ZFS Storage Appliance

Prerequisites and Monitoring Credentials

Monitoring credentials are required to monitor most entities using Cloud agents. Defining monitoring credentials is a **prerequisite** step for adding most entities to your monitoring service. Each entity that requires monitoring credentials must have these created or identified ahead of time. Credentials are then passed on to the Cloud agents using credentials JSON files. In addition to monitoring credentials, some entity types may require additional setup and configuration before they can be monitored by Infrastructure Monitoring. The following table provides links to prerequisite and monitoring credential information.

Entity Type	Entity
Hosts	<ul style="list-style-type: none"> • Host • Fujitsu Server • Intel/SPARC Computers
Virtual Servers	<ul style="list-style-type: none"> • Oracle VM Server for SPARC (LDBoms) • Docker Engine / Docker Container • XEN Virtual Platform / XEN Virtual Server
Oracle DB Systems Provides single-step discovery of the Oracle DB and all related entities, such as DB, PDB, Clusterware, and Listeners	<ul style="list-style-type: none"> • For more information about discovering Oracle DB Systems, see Add Oracle Database Systems.
Relational Databases	<ul style="list-style-type: none"> • Oracle Database • Oracle NoSQL • AWS-RDS Oracle DB • MySQL Database • Microsoft SQL Server
NoSQL Databases	<ul style="list-style-type: none"> • Apache Cassandra DB • MongoDB Database
Storage Management	<ul style="list-style-type: none"> • Oracle Flash Storage • Oracle Automatic Storage Management (ASM)
Java Application Servers	<ul style="list-style-type: none"> • Tomcat • JBOSS
Java EE Application Servers	<ul style="list-style-type: none"> • Oracle Service Bus • Oracle WebLogic Server (includes WebLogic Domain and WebLogic Cluster)
Web Application Servers	<ul style="list-style-type: none"> • Apache HTTP Server • Oracle HTTP Server (OHS) • Microsoft Internet Information Services (IIS)
Load Balancers	<ul style="list-style-type: none"> • Oracle Traffic Director (OTD) • F5 BIG-IP LTM
Physical Switches	<ul style="list-style-type: none"> • Arista Ethernet Switch • Cisco Ethernet (Catalyst) Switch • Cisco Nexus Ethernet Switch • Juniper Ethernet Switch • Oracle Infiniband Switch • Brocade Fibre Channel Switch • ES2 Ethernet Switch
Applications	<ul style="list-style-type: none"> • SCOM (System Center Operations Manager) • VMware vCenter • Docker Swarm • Oracle Access Manager (OAM) • Oracle Clusterware (CRS) • Oracle E-Business Suite (Workflow for Setting up the Environment) • Oracle VM Manager
Firewalls	<ul style="list-style-type: none"> • Juniper SRX Firewall • Arbor TMS/CP • Juniper Netscreen Firewall

Entity Type	Entity
Networking	<ul style="list-style-type: none"> Juniper MX Router F5 BIG-IP DNS
Other	<ul style="list-style-type: none"> Oracle Power Distribution Unit (PDU) Apache SOLR Hadoop Cluster Coherence Oracle Unified Directory(OUD) Oracle Internet Directory (OID) Oracle Identity Manager (OIM) Kubernetes Cluster Oracle GoldenGate

Locate the type of entity you wish to monitor. Then, follow the corresponding configuration steps described. Check also My Oracle Support *Infrastructure Monitoring Cloud Service Master Note (Doc ID 2195015.1)* for more release-specific information.

Enable Host Monitoring

Your hosts are automatically added as entities when a Cloud agent is installed. Monitoring of host entities, however, is disabled by default.

You can enable host monitoring directly from the Oracle Management Cloud console. Depending on the type of licensing, you may or may not see a **Licensing** option, so the procedures will differ.

To determine whether the **Licensing** option is available, from the Oracle Management Cloud console navigation pane, select **Administration** → **Entity Configuration**

Versions with the Licensing Option

1. Click **Licensing**.
2. Click **Select Entities** and choose the desired host.
3. Select **Standard Edition** or **Enterprise Edition**.
4. Click **Save**.

Versions without the Licensing Option

1. Click **Enable/Disable Services**.
2. Click **Select Entities**. The *Select Entities* dialog displays.
3. Select the desired host target from the list and click **Select** to close the dialog. The host appears in the Entity list. Ensure that Monitoring has been enabled for the entity.
4. If the Monitoring service is not already enabled, click **Enable Services**. The Enable Services dialog displays.
5. Toggle the Monitoring service on and click **Enable Services**.

If you want to enable host monitoring using the command line interface (OMCLI) and JSON files, the *capability* property must be set to *monitoring* as shown in the following procedure.

1. Edit the sample Host JSON file provided with the appropriate values for your hosts.
The sample JSON file you downloaded (`update_host_sample_1.14_and_on.json`)

- **name:** Your local host name used for your Cloud Agent installation.
- **type:** Your host type. The options are:
 - omc_host_linux
 - omc_host_solaris
 - omc_host_windows
 - omc_host_aix

The best way to determine the correct values of the entity **name** and **type** is to query the agent on your host. For example, run the following:

```
./omcli status_entity agent
Oracle Management Cloud Agent
Copyright (c) 1996, 2018 Oracle Corporation. All rights reserved.
  Lama.host1.example.com:3872      : AGENT:entity
successfully added
  omc_host_linux.host1.example.com : AGENT:entity
successfully added
```

In this case, your JSON file will become:

```
{
  "entities":
  [
    {
      "name": "host1.example.com",
      "type": "omc_host_linux",
      "properties":{
        "capability": {
          "displayName": "capability",
          "value": "monitoring"
        }
      }
    }
  ]
}
```

2. Update your host entity. From the agent installation directory (for example, on a UNIX system, <AGENT_BASE_DIR>/agent_inst/bin) run:

```
./omcli update_entity agent <Your Host JSON input file name>.json
```

For example,

```
./omcli update_entity agent update_host_host1.json
```

If you have an HA configuration (a virtual host with two or more physical hosts configured with failover software) note the following:

- Your Cloud agents must be installed on all hosts (virtual and physical hosts).
- The Cloud agents on your *physical* hosts will monitor the host entities. Therefore, the steps listed above to enable host monitoring must be performed on the *physical* hosts.

- The *other* entities you want monitored must be added using the Cloud agent on the *virtual* host.

Add Entities from the Console

For entities that are monitored by Cloud Agents, you can alternatively add them directly from the Oracle Management Cloud **Add Entities** page. This greatly simplifies the entity addition process.

When adding an entity from the UI, the entity-specific properties are submitted as an Oracle Management Cloud discovery job. For more information about specific entity attributes and properties, see [Entity Attributes and Properties](#). The new entity will be added upon successful completion of the job.

Note:

Not all agent-monitored entity types can be added using the Oracle Management Cloud console. For entity types not available as a selectable option, you will need to add them manually using the Oracle Management Cloud command line interface (*omcli*). See [Add Entities Using JSON Files](#).

Adding an Entity

To add an entity from the Add Entity page.

1. From the Management Cloud main menu, select **Administration, Discovery**, and then **Add Entity**. The Add Entity page displays.
2. Select an **Entity Type**. Property and monitoring credential fields specific to the selected entity type are displayed as shown in the following graphic.

Note:

After a fresh Cloud agent installation, agent configuration information first needs to be collected before it can be selected from the Add Entity UI. Wait five minutes after a Cloud agent installation before selecting it from the **Cloud Agent** drop-down list.

By default, the discovery *Job Name* consists of the entity type and timestamp. You can change this name to something more intuitive, if desired.

Optionally, you can create tags that define additional relationships between entities. These relationships will help search and group these entities in Oracle Management Cloud. For example, you may want to use the same tag for all entities that are physically in the same location, or entities that are part of the same custom logical group. The *Tag all members* option applies to composite entities and allows you to specify the same tag to all members discovered under this entity.

The tags can also be specified in the entity JSON file. For information about tags, see [Define Associations Between Entities](#).

 **Note:**

If you cannot find the desired entity type from the drop-down list, you may have to add the entity using omcli. See [Add Entities Using JSON Files](#).

3. Enter the requisite properties, monitoring credentials, and tags if desired. The entity properties shown on this page mirror those found in the entity type's JSON file. See [Entity Attributes and Properties](#) for more information about entity type properties.
4. Click **Add Entity**. You are returned to the *Discovery Job Status* page. The entity addition job name specified in step 3 will appear in the table.

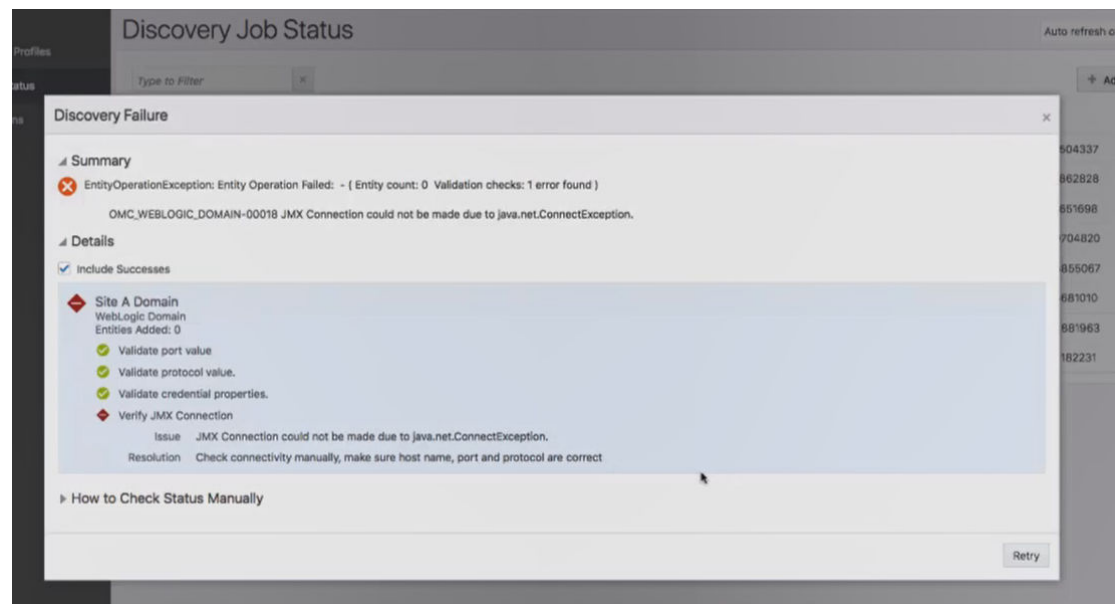
From the Discovery Job Status page, you can view the current status of entity addition attempts. Entity status can be one of the following:

- **Success** - This is a terminal state where the discovery job is a success.
- **Submitted** - This is the initial state when the discovery job has just been submitted.
- **In Progress** - This is the state when the discovery is on-going.
- **Failure** - This is a terminal state when the discovery job has failed.
- **Success with warnings** - This is a terminal state where entity discovery has succeeded but there are some warnings.

For entity discovery jobs with a *Failed* or *Warning* status, **Retry** appears in the Action column. Clicking **Retry** will open up the Add Entity page populated with parameters (except monitoring credentials) that were used when the entity was originally added. This allows you to quickly make any corrections and resubmit the entity discovery job without having to enter all the parameters.

Prerequisite Checks

When the discovery job is run, prerequisite checks are automatically performed for certain entity types. You can view the output of the prerequisite checks by clicking on the discovery job status for that entity. The output contains valuable information that can help you diagnose discovery job failures.



The screenshot shows the 'Discovery Job Status' page with a 'Discovery Failure' dialog box open. The dialog box has a 'Summary' section with a red 'X' icon and the text: 'EntityOperationException: Entity Operation Failed: - { Entity count: 0 Validation checks: 1 error found }'. Below this is the specific error: 'OMC_WEBLOGIC_DOMAIN-00018 JMX Connection could not be made due to java.net.ConnectException.'. The 'Details' section is expanded, showing a list of checks for 'Site A Domain' (WebLogic Domain) with 'Entities Added: 0'. The checks are: 'Validate port value' (passed), 'Validate protocol value' (passed), 'Validate credential properties' (passed), and 'Verify JMX Connection' (failed). The failed check details show the issue: 'JMX Connection could not be made due to java.net.ConnectException.' and the resolution: 'Check connectivity manually, make sure host name, port and protocol are correct'. A 'Retry' button is visible at the bottom right of the dialog box.

Composite Entities

Composite entities consist of multiple separate child entities. For example, a database system entity can consist of databases, listeners, pluggable databases, etc. When the discovery job is run to add a composite entity, it's possible that only some of the child entities will not be added successfully. When this happens, we recommend that you follow this procedure for each entity that was not successfully added:

- Perform a **Retry** for the entity
- If you're using OMCLI, run `omcli update_entity` followed by `omcli refresh_entity`.

Forcing Discovery

Under certain circumstances, you may want a discovery job to proceed regardless of what issues are raised by the prerequisite checks. For example, when installing a composite entity such as a database system, even though prerequisite validation issues occur when trying to discover a net listener, you may only care about discovering the RAC databases, ASM, and other select database system child entities rather than stopping the discovery process for the entire database system. To do this, you can force the discovery job to proceed via the *force* option.

When the *force* option is used, you tell the discovery job to report all issues that occur when the prerequisite validation checks are run, but ignore the check recommendations and to go ahead with the discovery attempt.

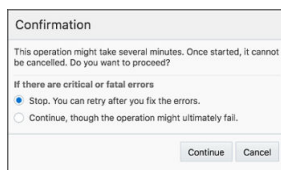
There are two ways to specify the force option:

From OMCLI:

Specify the `-force` option when executing the `add_entity` or `update_entity` verb.

From the Oracle Management Cloud Console:

1. Click **Add Entity** (or **Retry**). The **Confirmation** popup displays.



2. Select **Continue, though the operation might ultimately fail**.
3. Click **Continue**.

Define Associations Between Entities

By defining associations between entities you are monitoring, you can make Oracle Infrastructure Monitoring (and OMC in general) reflect the topological, as well as operational dependencies between them. For example, you create an association between two entities: Apache Tomcat and MySQL database. By creating a *tagged* relationship between the two entities, Oracle Cloud can take advantage of these relationships by having Oracle Application Performance Monitoring display the relationship in a topology map, help you understand where some of the load being

placed on the MySQL database is coming from, or let you see what impact bringing down the database will have.

Add Associations using the Entity Associations UI

You can add/manage entity associations directly from the Entity Association page.

1. From the Oracle Management console main menu, select **Administration**—>**Discovery** —>**Entity Associations**.
2. Click **Select the Source Entity**. The Select Entities dialog displays.
3. Select the source entity and click **Select**. Default associations appear in the table.
4. Click **Add Associations**. The Add Associations dialog displays.
5. Select the desired **Association Type** from the drop-down menu: *Uses*, *Contains*, or *Monitored by*.
6. Click **Add Destination Entities**. The Select Entities dialog displays.
7. Choose the desired Destination entities and click **Select**.
8. On the Add Associations dialog, check *I want to add these associations* box and click **Save**. The newly defined association appears in the table.

Add Associations while Adding Entities

You can also add association tags when adding entities via the Oracle Management Cloud console. The Tags region of the Add Entities UI lets you specify tags directly without any coding. For more information about adding entities from the console while adding an entity, see [Add Entities from the Console](#).

Discover Cloud Services

Oracle Management Cloud's Monitoring Cloud services capability gives you an easy way to obtain monitoring data from Cloud service entities such as Amazon Web Services or Microsoft Azure. By defining a monitoring *Discovery Profile* that is used to access the desired service, you can monitor Cloud services with minimal setup.

About Monitoring Cloud Services

To monitor a Cloud service, you first create a Cloud Discovery Profile. This profile defines Cloud service account information required to discover services and monitor them as entities. Monitoring of Cloud services automatically starts as soon as the services are discovered. Every 15 minutes, Oracle Management Cloud automatically checks for new services and also automatically polls metric data from your monitored services. Once the Cloud services are discovered, and status and performance metrics have been collected, monitoring features such as alerting and notifications will be automatically applied to the Cloud services.

Adding a Cloud Service

You add Cloud services by defining a *Cloud Discovery Profile*.

1. Navigate to the Cloud Discovery Profiles page (Administration—>Discovery—>Cloud Discovery Profiles).
2. Click **Add Profile**. The **Add Discovery Profile** page displays.

The screenshot shows the 'Add Discovery Profile' page in Oracle Management Cloud. The interface includes a sidebar with 'Discovery', 'Add Entity', 'Cloud Discovery Profiles', and 'Discovery Job Status'. The main content area has a title 'Add Discovery Profile' and a sub-header 'Oracle Management Cloud will discover cloud services associated with a single account. The discovery process might take several minutes. Once started, it cannot be cancelled.' There are 'Start Discovery' and 'Cancel' buttons. The form contains the following fields:

- * Profile Name: Opc
- * Cloud Service Provider: Oracle Cloud
- Regions and Services:

Region	Services
US	Compute Cloud Service X
- Credentials:
 - Existing Credentials
 - New Credentials
 - * Credential Name: [text input]
 - * Identity Domain: [text input]
 - * User Name: [text input]
 - * Password: [text input]

3. Enter a **Profile Name** and select a **Cloud Service Provider**. Based on the *Cloud Service Provider* you will create a profile that encapsulates all information required to connect to the Cloud vendor.

 **Note:**

Required discovery profile information changes according to the *Cloud Service Provider* you select. See the following section for discovery information required for each service provider.

4. Click **Start Discovery**.

Cloud Service Entity Type Discovery Information

Oracle Cloud

- Regions and Services
 - *Region*: Region in which your services are enabled. US or Europe
 - *Service*: Services enrolled in Oracle Public Cloud that are to be monitored by Oracle Management Cloud.
- Credentials
 - *Credential Name*: Any name for the credentials account.
 - *Identity Domain*: If you are using the traditional account, specify the Identity Domain. If you are using an Identity Cloud Service (IDCS)-based account, specify the Identity Service ID. This would be of the form `idcs-<GUID>`.
 - *Username*: Username from Oracle Public Cloud.
 - *Password*: Password from Oracle Public Cloud.

Using Cloud Discovery Profiles with Single Sign-on

 **Note:**

For monitoring via cloud profiles, only Oracle Compute is supported. Database Cloud Service and Java Cloud Service can be discovered via Cloud profiles but they are only supported for Compliance Service. To monitor (unmanaged) DBCS and JCS, you should use the cloud agent to monitor it like an Oracle database and WebLogic Server.

If IDCS single sign-on has been enabled, you will need to perform the following procedures in order to enable monitoring using these IDCS-based accounts.

Find the Identity Domain to be provided for adding Oracle Public Cloud (OPC) profiles for an IDCS-based account:

1. Log in and navigate to the **MyServices** page <https://myservices-<tenant id>.console.oraclecloud.com/mycloud/cloudportal/dashboard>
2. Click **Customize Dashboard** and select **Identity Cloud** from the drop-down list.
3. In the overview page, search for **Identity Service Id**. This corresponds to the IDCS GUID that should be used when creating OPC cloud profiles.

For an IDCS-based user account, grant the *Monitoring_ApiAccess* privilege to the user:

1. Log in and navigate to the MyServices page <https://myservices-<tenant id>.console.oraclecloud.com/mycloud/cloudportal/dashboard>
2. At the upper-right corner, click **Users**.
3. Navigate to the tab **Groups**.
4. Click **Add** and create a group with name *Monitoring_ApiAccess* (if the group does not exist already)
5. Click on the created group.
6. Navigate to the tab **Users**.
7. Click **Add To Group** at the right side. This lists all the existing users. Select the user for which you want to grant access to this group and click **Add**.

For a traditional account, perform the following steps to grant *Monitoring_ApiAccess* privilege to the user:

1. Connect to <https://myservices.us.oraclecloud.com/mycloud/faces/cloudHome.jspx>
2. Scroll down the page and click **MyServices**.
3. Click **Users**. <https://myservices.us.oraclecloud.com/mycloud/faces/security.jspx>
4. Create a Custom Role as shown in <http://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/ramoc/QuickStart.html>. with the Role name *Monitoring_ApiAccess* and the display name *Monitoring API Access*.
5. After creating the *Role* assign it to the *user*..

 **Note:**

The role needs to be created by the identity domain administrator.

Amazon Web Services

Oracle Management Cloud executes AWS functions to monitor AWS Entities. AWS users must have the permissions shown in the following table for discovery and monitoring of AWS services.

 **Note:**

The AWS permissions should be assigned to the user or the group to which the user belongs. Role-based access is currently not supported.

AWS Service	AWS Entity	Function	Required for
DynamoDB	omc_aws_dynamodb_table	ListTables	Discovery
EC2	omc_aws_ec2_instance	DescribeInstances	Discovery
	omc_aws_ebs	DescribeVolumes	Discovery
	omc_aws_elastic_ip	DescribeAddresses	Discovery
	omc_aws_security_group	DescribeSecurityGroups	Discovery
	omc_aws_customer_gateway	DescribeCustomerGateways	Discovery
	omc_aws_internet_gateway	DescribeInternetGateways	Discovery
	omc_aws_route_table	DescribeRouteTables	Discovery
	omc_aws_subnet	DescribeSubnets	Discovery
	omc_aws_vpc	DescribeVpcs	Discovery
	omc_aws_vpn_connection	DescribeVpnConnections	Discovery
Elastic Load Balancer	omc_aws_elb_instance	DescribeLoadBalancers	Discovery
	omc_aws_elb_application_instance	DescribeLoadBalancers	Discovery
Lambda	omc_aws_lambda_function	ListFunctions	Discovery
RDS	omc_aws_rds_instance	DescribeDBInstances	Discovery
Redshift	omc_aws_redshift_cluster	DescribeClusters	Discovery
S3	omc_aws_s3_bucket	ListAllMyBuckets	Discovery
SNS	omc_aws_sns_topic	ListTopics	Discovery
SQS	omc_aws_sqs_queue	ListQueues	Discovery
CloudWatch	.	GetMetricStatistics	Performance collection for all the entities.

- *AWS Account Number*: Amazon Identity and Access Management (IAM) user name.
- *Regions and Services*

Most Amazon Web Services offer a regional endpoint to make your requests. An endpoint is a URL that is the entry point for a web service. For example, <https://>

dynamodb.us-west-2.amazonaws.com is an entry point for the Amazon DynamoDB service.

- Credentials
 - Credential Name: Any name for the credentials account.
 - AWS User Access Key: Access keys consist of an access key ID (Example: AKIAIOSFODNN7EXAMPLE)
 - AWS User Secret Key: A secret access key consisting of a secret key ID (Example, wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY)

Microsoft Azure

- *Azure Subscription ID*: An Azure GUID (subscription ID).
- Regions and Services
 - *Region*: All Regions - Microsoft Azure endpoints are global. Because selection of resources to be monitored by region is not supported,, all resources for given subscription ID will be monitored.
 - *Services*: Currently, only monitoring of Azure VM instances is supported.

- Credentials


Azure Monitoring is done through Azure Resource Manager and Azure Monitor APIs using an App account within the specified Azure Active Directory and has read permission for the specified subscription(s). The App account can have read permission for multiple subscriptions; therefore credentials can be reused.

For information on setting up Azure monitoring from Oracle Management Cloud, see [Microsoft Azure](#).

- *Credential Name*: Name given to credentials.
- *Microsoft Active Directory ID*: Azure Active Directory → Properties → Directory ID (Example: cafe8c3d-c91a-4405-a43b-01efee6d2fbc)
- *Microsoft Active Directory Application ID*: Azure Active Directory → App registrations → App / APPLICATION ID (Example: cafef0f5-f431-4c8b-9ee3-22524407ce69)
- *Microsoft Active Directory Application Key*: Displayed on the registration of the App in the Azure Active Directory

Supported Cloud Entity Types

Cloud Vendor	Cloud Service	Monitored by REST APIs	Monitored by Cloud Agent
Oracle Cloud	Compute (General Purpose and Dedicated Compute) (OCI Classic only)	Yes	Yes (agent is local to Compute)

 **Note:** To monitor Compute via REST APIs, you must have t

Cloud Vendor	Cloud Service	Monitored by REST APIs	Monitored by Cloud Agent
--------------	---------------	------------------------	--------------------------

h
e
M
o
n
i
t
o
r
i
n
g
-
a
p
i
A
c
c
e
s
s
r
o
l
e
. T
h
e
r
e
a
r
e
t
w
o
w
a
y
s
t
o
p
e
r
f
o
r
m
t
h
i

Cloud Vendor	Cloud Service	Monitored by REST APIs	Monitored by Cloud Agent
--------------	---------------	------------------------	--------------------------

s
a
c
t
i
o
n
d
e
p
e
n
d
i
n
g
o
n
w
h
e
t
h
e
r
y
o
u
a
r
e
u
s
i
n
g
a
t
t
r
a
d
i
t
i
o
n
a
l
a
c
c
o
u
n

Cloud Vendor	Cloud Service	Monitored by REST APIs	Monitored by Cloud Agent
--------------	---------------	------------------------	--------------------------

t
o
r
a
l
D
C
S
-
b
a
s
e
d
a
c
c
o
u
n
t
.
F
o
r
m
o
r
e
i
n
f
o
r
m
a
t
i
o
n
,
s
e
e
[Q
u
i
c
k
S
t
a
r
t](#)
:

Cloud Vendor	Cloud Service	Monitored by REST APIs	Monitored by Cloud Agent
			O b t a i n A c c o u n t I n f o r m a t i o n
.	Database Cloud Service (11g and 12c)	No	Yes (agent can be local or remote)
.	Java Cloud Service (WebLogic Server 11g and 12c)	No	Yes (agent can be remote)
.	Exadata Cloud Service	No	Yes: Oracle DB, Listener, Host/VM components
Amazon	Elastic Compute Cloud (EC2)	Yes	Yes (agent is local)
.	Relational Database (RDS) - Oracle	Yes	Yes
.	RDS (all database engines)	Yes	No
.	Simple Storage Service (S3)	Yes	No
.	Elastic Block Store (EBS)	Yes	No
.	Redshift	Yes	No
.	Elastic Load Balancer (ELB) – Classic Load Balancer and Application Load Balancer	Yes	No
.	Lambda	Yes	No

Cloud Vendor	Cloud Service	Monitored by REST APIs	Monitored by Cloud Agent
.	Simple Notification Service (SNS)	Yes	No
.	Simple Queue Service SQS	Yes	No
Microsoft Azure	Virtual Machines	Yes	No
.	Logic Application Service	Yes	No
.	VM Scale Set	Yes	No
.	API Application Service	Yes	No
.	Application Service Plan	Yes	No
.	Application Gateway Service	Yes	No
.	Event Hub Namespace Service	Yes	No
.	Functions Application Service	Yes	No
.	Mobile Application Service	Yes	No
.	Web Application Service	Yes	No
.	SQL Database	Yes	No
.	SQL Data Warehouse	Yes	No

Support for OCI Compute

OCI Comput is currently supported and monitored like a host entity using the cloud agent. Once you deploy the cloud agent, the underlying host should be discovered and monitored like an host entity.

Support for Autonomous Database in OCI

For information about Autonomous Database in OCI, see Discover Autonomous Databases in *Using Oracle Database Management for Autonomous Databases*.

Add Entities Using JSON Files

In order to monitor various entities, you need to first add them to Oracle Infrastructure Monitoring. Adding new entities to your service includes the tasks listed below.

Note:

You will need to add *each* entity and its corresponding credentials (if applicable) to its local monitoring agent (Cloud Agent). In some cases, remote monitoring is supported, see My Oracle Support *Infrastructure Monitoring Cloud Service Master Note (Doc ID 2195015.1)* for more release-specific information.



Note:

Before you begin, ensure that all required agent deployment steps have been performed. These steps are part of the initial setup of your service, see *Install Cloud Agents in Installing and Managing Oracle Management Cloud Agents*.

Oracle by Example

For examples on adding entities, take a look at the following tutorials:

- [Add a MongoDB Entity](#)
- [Add a WebLogic Server Entity to Oracle Log Analytics and Later Oracle Infrastructure Monitoring](#)

1. Identify the Entity Types You Want to Monitor

You add entities to Oracle Infrastructure Monitoring directly through the UI, or by adding their respective JSON files to the system. The following table lists entities that the Cloud Agent can monitor for the Oracle Infrastructure Monitoring Service in the current release. Make a note of the entity type(s) you will want to monitor with your service.

In addition to being able to monitor conventional entity types, Infrastructure Monitoring also allows you to monitor Cloud services for third-party vendors that provide Cloud service REST APIs. The following table lists the current

2. Set Up Monitoring Credentials

Monitoring credentials are required to monitor some of the entities using Oracle Infrastructure Monitoring. To locate your entities and set up monitoring credentials, see [Prerequisites and Monitoring Credentials](#).

3. Downloading and Customizing JSON Files

For monitoring, you must create two types of JSON files that contain information about the entities to be monitored:

1. An **entity definition JSON file** for each entity type you're adding.
2. A corresponding **credentials JSON file** for each entity you're adding, if credentials are required to monitor this entity.

To download and customize the JSON files that correspond to your entities, see [Download and Customize Oracle Infrastructure Monitoring JSONs](#).

For information on the various properties and attributes associated with each entity, see [Entity Attributes and Properties](#).

Encrypting the Credentials JSON File

For security, you can use GNU Privacy Guard (GPG) to encrypt text-based credential JSON files using asymmetric keypairs (public and private).

Prerequisites:

- Ensure both the entities JSON file and credential JSON file have a *.json* extension.

- GPG keys have been added to the Linux host in order to convert **creds.json** into **creds.json.gpg** (encrypted GPG file).

For more information on adding GPG keys, see: <https://www.cyberciti.biz/tips/linux-how-to-create-our-own-gnupg-privatepublic-key.html>

To convert the JSON files, run the following command::

```
gpg -c <path and name of credential json file>
```

Enter the password (provided while adding GPG keys) twice.

The credential file is converted to GPG format. For example, *Db_creds.json.gpg*.

- Ensure the agent is up and running.

Procedure

Note:

The example used in this procedure shows how to encrypt credentials for a database entity using GPG.

1. Add the entity.

```
./omcli add_entity agent add_db.json -credential_file add_db_creds.json.gpg
-encryption_method_gpg
```

Enter the passphrase that was provided when you added the GPG keys to the host.

2. Verify the status of the newly added entity.

```
./omcli status_entity agent add_db.json
Oracle Management Cloud Agent
Copyright (c) 1996, 2016 Oracle Corporation. All rights reserved.
omc_oracle_db.OracleDb11 : AGENT:entity fully monitored
```

3. Verify that the newly added target appears when you list targets on the host.

```
./omcli config agent listtargets
Oracle Management Cloud Agent
Copyright (c) 1996, 2016 Oracle Corporation. All rights reserved.
[myhost.com, omc_host_linux]
[myhost.com:1839, Lama]
[OracleDb11_sys, omc_oracle_db_system]
[OracleDb11, omc_oracle_db]
[OracleDb11/orcl12c, omc_oracle_db_instance]
```

4. Verify that the target and credentials are encrypted in target.xml. *<Agent_Home>/agent_inst/sysman/emd/targets.xml*

```
<Property NAME="DBPassword" VALUE="{ENC S}
{AES-128}3DC4D610690287A389D913ECEA40531CF12DAFAD13940100"
ENCRYPTED="TRUE"/>
<Property NAME="DBRole" VALUE="{ENC S}
{AES-128}4D98B5141164AD038C3634C1C8CED0BC56238B3173A0" ENCRYPTED="TRUE"/
```

```
>  
<Property NAME="DBUserName" VALUE="{ENC S}  
{AES-128}F4D7D350906F9778E45880A085AF1D4F164F7B3264A49C59E466FAE09B"  
ENCRYPTED="TRUE"/>
```

4. Adding Entities to Your Service

Run the following command from the agent installation directory, for example, on a UNIX system this directory is (<AGENT_BASE_DIR>/agent_inst/bin) to add each entity to your monitoring service:

Note:

When specifying the full path to a JSON file, make sure there are no spaces as these will cause the `oemcli add_entity` command to fail.

```
oemcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

In this command:

- **DEFINITION_FILE** is the entity definition JSON file that contains the details about the entity to be added. This file does not contain any actual credentials but it must contain a reference to credentials specified in the credentials file.
- `-credential_file` is the credentials file parameter.
- **CREDENTIAL_FILE** is the name of the credentials JSON file. You must define corresponding credentials for each entity that requires them, even if multiple entities happen to have the same monitoring credentials. Some entities may not require credentials for monitoring, in which case the `-credential_file` parameter can be omitted.
- `-encryption_method_gpg` is an optional parameter; if specified, this option indicates that the file is encrypted using GNU Privacy Guard symmetric encryption.

GNU Privacy Guard (GPG) is an encryption software that is OpenPGP (RFC 4880) compliant. You can use GPG to encrypt and decrypt files that contain sensitive data. To avoid saving sensitive information such as passwords in clear text, you can first encrypt the credential JSON file with GPG and then use the encrypted JSON file along with the `-encryption_method_gpg` flag when adding your entity to the monitoring service.

For example, to add a database entity to an agent running on a UNIX system, run:

```
./oemcli add_entity agent omc_oracle_db_prodl.json -credential_file  
omc_oracle_db_creds.json
```

```
Oracle Management Cloud Agent  
Copyright (c) 1996, 2016 Oracle Corporations. All rights reserved.  
Operation Succeeded; Accepted 1 of 1 entities for processing.
```

If you have an HA configuration (a virtual host with 2 or more physical hosts configured with failover software) note the following:

- Your Cloud Agents must be installed on all hosts (virtual and physical hosts).
- The Cloud Agents on your *physical* hosts will monitor the host entities. Therefore, the steps to enable host monitoring must be performed on the *physical* hosts, see [Monitoring Credentials](#).
- The entities you want monitored must be added using the Cloud Agent on the *virtual* host.

5. Verifying Added Entities

Verify your entity addition by running the following command from the same agent directory (<AGENT_BASE_DIR>/agent_inst/bin):

```
omcli status_entity agent DEFINITION_FILE
```

where **DEFINITION_FILE** is the entity definition JSON file.

When the addition is complete, the verification will indicate that the entity is fully monitored. For example, if running on a UNIX host, verify as follows:

```
./omcli status_entity agent omc_oracle_db_prod1.json
```

```
Oracle Management Cloud Agent  
Copyright (c) 1996, 2019 Oracle Corporation. All rights reserved.  
omc_oracle_db.prod1 : AGENT : entity fully monitored
```

Navigate to the Enterprise Summary dashboard and note the new entity you added. Depending on your network latency and other load factors, allow a few minutes for this process to complete.

Troubleshooting

View Prerequisite Checks

If there are issues when adding an entity, you can obtain more information about the entity addition process by using the `-verbose` option when performing the verification. This option displays the results of the prerequisite checks which are run automatically during the entity discovery process. For each prerequisite check, the status and, if there is an error, the issue and associated resolution is shown.

```
omcli status_entity agent DEFINITION_FILE -verbose
```

```
./omcli status_entity agent apache_entity.json -verbose  
Oracle Management Cloud Agent  
Copyright (c) 1996, 2019 Oracle Corporation. All rights reserved.  
omc_generic_apache.TestApache2 : AGENT:EntityOperationException: Entity  
Operation Failed: - (Entity count: 0 Validaiton checks: 2 errors found)  
      OMC_GENERIC_APACHE-00002 Invalid value for property Is Remote  
(omc_is_remote). Provided value [ yres ] expected value is Yes or No.  
Details:  
CHECK 1      : Validate input property - Listen Port (omc_listen_port)
```

```

Status      : SUCCESS
Check 2     : Validate input property -Is Remote (omc_is_remote)
Status      : ERROR
Issue       : OMC_GENERIC_APACHE-00002 Invalid value for property Is
Remote
              (omc_is_remote).Provided value [ yres ] expected value
is Yes or No.
Resolution  : Provide expected value for property Is Remote
(omc_is_remote) - 'Yes/No' .
CHECK 3     : Validate if /server-status page [ http://
host.company.com:54123/server-status ] is accessible
Status      : ERROR
Issue       : com.sun.jersey.api.client.ClientHandlerException:
java.net.ConnectException:
              Connection refused (Connection refused)
Resolution  : Enable access to Apache status page [ http://
host.company.com:54123/server-status ] for host [ host.company.com ].
Update (or add) the block that starts with <Location /server-status>
in httpd.conf file. Graceful restart of Apache HTTP Server is required.

```

Check the Cloud Agent Log Files

For additional troubleshooting information, check the Cloud Agent logs. Entities discovery information is logged in the agent logs, on the hosts where agents are installed. For example, on UNIX hosts these are located in the `<AGENT_BASE_DIR>/sysman/log/` directory:

While monitoring an Oracle Database, if you encounter metrics collection errors, it is possible that the monitoring password has expired. To reset the password and re-enable metrics collections:

- gcagent_sdk.trc
- gcagent.log

1. Log in to the monitored database entity and alter the login attempts limit:

```

SQL> ALTER PROFILE DEFAULT LIMIT FAILED_LOGIN_ATTEMPTS UNLIMITED;
SQL> Alter user <monitoring user> account unlock;

```

2. On the agent host, update the entity monitoring password. For example, on a UNIX host:

```

./omcli update_entity <the original host entity JSON file>

```

3. Return to the database entity and reset the login attempts limit to the recommended value 5:

```

SQL> ALTER PROFILE DEFAULT LIMIT FAILED_LOGIN_ATTEMPTS 5;

```

Forcing Entity Discovery

Under certain circumstances, you may want a discovery job to proceed regardless of what issues are raised by the prerequisite checks. For example, when installing a composite entity such as a database system, even though prerequisite validation

issues occur when trying to discover a net listener, you may only care about discovering the RAC databases, ASM, and other select database system child entities rather than stopping the discovery process for the entire database system. To do this, you can force the discovery job to proceed via the force option.

When the force option is used, you tell the discovery job to report all issues that occur when the prerequisite validation checks are run, but ignore the check recommendations and to go ahead with the discovery attempt.

Specify the `-force` option when executing the `omcli add_entity` or `update_entity` verb.

```
./omcli update_entity <entity definition json file> -force
```

Download and Customize Oracle Infrastructure Monitoring JSONs

Download the set of JSON files provided for the Oracle Infrastructure Monitoring, included in the `Infrastructure Monitoring` sub-folder of this zip file:

[Sample JSON files \(zip file\)](#)

General Guidelines For Editing JSON Files

- Identify the JSON files you will be working with. The name of each file is based on the entity internal names. For example, the file `omc_mysql_db_sample.json` is the JSON entity definition file for an entity whose internal name is `omc_mysql_db` (My SQL Database) and its corresponding credentials file is `omc_mysql_creds.json`.
- **IMPORTANT!** In entity definition files, change the values indicated in the angle brackets (`< >`) and remove the angle brackets. **DO NOT** change any other values.
- **IMPORTANT!** In credentials files, change the values indicated in the square brackets (`[]`) and keep the square brackets. **DO NOT** change any other values.
- Consider using a JSON validation tool when you are ready to save your files.
- Check My Oracle Support *Infrastructure Monitoring Cloud Service Master Note (Doc ID 2195015.1)* for certified entity versions and more release-specific information.

JSON Files Description

The **entity definition JSON file** helps you specify the type of entities that you want to monitor with your service and includes some of the required parameters for a successful discovery into your service. The entity definition files include the following:

- Name and type of entity, as well as other entity-specific information, such as entity connection information.

 **Note:**

Entity naming restrictions: Entity names cannot be longer than 256 characters. You can use any alphabetic, numeric or the following special characters:

- underscore “_”
- dash “-”
- at “@”
- hashtag “#”
- colon “:”
- forward slash “/”
- equal “=”
- period “.”

- References to a credentials unique identifier specified in a corresponding credentials file.

An entity definition file is accompanied by a corresponding **credentials JSON file**, if that entity requires monitoring credentials. The credentials JSON file contains the credentials (for example, a user name and password pair) for that entity that allows the service to perform monitoring tasks. You can list these credentials directly in your JSON file or include them in a separate local file.

A credentials file follows this format:

```
{"credentials":  
[  
  {"id":"id1", "name":"credName1", "credType":"type1",  
   "properties":[{"name":"prop1", "value":"CLEAR[value1]"},  
                 {"name":"prop2", "value":"FILE[/tmp/filename]}]}  
]}
```

This sample format includes:

- A credential with an id (`id1`). This credential must match the credential reference in the entity definition and must be unique.
- A name (`credName1`), which can be any name that you specify to distinguish your credentials.
- A credential type, from a predefined set of known types (for example, `DBCreds` for databases).
- A property name (`prop1`) whose value (`value1`) is specified in clear text.
- A property (`prop2`) whose value is the content of the file `/tmp/filename`. See the table below for details on these variables.

Examples

This is an example of an Oracle Database entity definition JSON file named `omc_oracle_db_prodl.json`. This file specifies the details for a database with SID

PROD1, hosted on myhost.example.com and listening on port 1521. This file points to a credentials unique ID, SQLCreds, that must be defined in a credentials file. Note the fields in bold are the only ones that needed to be customized and the angled brackets (< >) are removed.

```
{
  "entities": [{
    "name": "oracle_PROD1",
    "type": "omc_oracle_db",
    "displayName": "West Coast Financials Production (Oracle)",
    "timezoneRegion": "America/Los_Angeles",
    "credentialRefs": ["SQLCreds"],
    "properties": {
      "host_name": {
        "displayName": "host_name",
        "value": "myhost.example.com "
      },
      "port": {
        "displayName": "port",
        "value": "1521"
      },
      "sid": {
        "displayName": "sid",
        "value": "PROD1"
      },
      "capability": {
        "displayName": "capability",
        "value": "monitoring"
      }
    }
  ]
}
```

The following is an example of the required credentials file, omc_oracle_db_creds.json that corresponds to the Oracle Database entity definition JSON file above. The credentials JSON file defines the unique credentials ID, SQLCreds, referenced in the entity definition file and specifies the connection to the Oracle Database instance as user name moncs, password moncs with a role Normal.

Note the square brackets are not removed.

```
{
  "credentials" : [
    {
      "id" : "SQLCreds",
      "name" : "SQLCreds",
      "credType" : "DBCreds",
      "properties" : [
        {
          "name" : "DBUserName",
          "value" : "CLEAR[moncs]"
        },
        {
          "name" : "DBPassword",
          "value" : "CLEAR[moncs]"
        },
        {
          "name" : "DBRole",
```

```
        "value" : "CLEAR[Normal]"  
    }  
  }  
}
```

Use Tags to Define Associations Between Entities

You can define entity associations at the time you first add entities for monitoring. By adding one of two special tags within an entity JSON file, you define the entity as either a *source* entity or a *destination* entity. This source-destination pairing links two entities together--from a source entity to a destination entity.

The following two tags are used to define either a source or destination entity:

- `assoc_source` (Source Entity Marker tag is used for tagging source entities)
- `assoc_dest` (Destination Entity Marker tag is used for tagging destination entities.)

Tag Syntax:

```
assoc_source:<Association Hint>
```

```
assoc_dest:<Association Hint>
```

Once added, the tags will be visible on an entity's home page.

where *Association Hint* is the unique identifier used to associate two entities. The *Association Hint* used for the source entity must match the one specified for the destination entity.

You can specify up to 50 tags per entity. For each tag *key:value* pair, you can specify up to 128 characters for the key and 256 characters for the value.

Usage Example

You have an environment with five WebLogic servers and two databases. For the five WebLogic servers, the following tag is used to mark them as source entities:

```
assoc_source:my_prod_dbs.
```

For the two databases, the following tag is used to mark them as destination entities:

```
assoc_dest:my_prod_dbs.
```

These tags will create associations between each WebLogic server and the two databases.



Note:

The *Association Hint* must be unique for each entity type.

Once you've defined the entity associations and added the entities, you can view them in the topology map.

Add Association Tags to Entity JSON Files

As discussed above, you create the association when you add a new entity. The following JSON file examples illustrate how the tags are implemented in the entity

JSON files. The examples used in the following steps create a tagged relationship between a host and a database.

Step 1: Define a Source

The first step is to define the association source. The following example shows how the source tag is implemented in a host JSON.

```
{
  "entities":
  [
    {
      "name": "<Your local host name that was used for agent
install>",
      "type": "<Your host Type>",
      "properties":{
        "capability": {
          "displayName": "capability",
          "value": "monitoring"
        }
      },
      "tags": {
        "assoc_source:tj_host" : ""
      }
    }
  ]
}
```

Step 2: Define an association destination.

To define an association destination, you add a tags section to the destination entity's JSON file using the *assoc_dest* tag to identify the entity as a destination. The following example shows how a tag section is added to a database destination JSON for the host shown in the previous step.

Example: Database JSON Destination Tag

The following example shows the database destination is associated with the source host entity named "tj_host".

```
{
  "entities":[
    {
      "name":"OracleDb_Tag",
      "type":"omc_oracle_db",
      "displayName":"OracleDb",
      "timezoneRegion":"PDT",
      "credentialRefs":[
        "SQLCreds"
      ],
      "properties":{
        "host_name":{
          "displayName":"dummy",
          "value":"abcde.myco.com"
        }
      },
    }
  ]
}
```

```
    "port":{
      "displayName":"Port",
      "value":"15212"
    },
    "sid":{
      "displayName":"SID",
      "value":"db1212"
    },
    "capability":{
      "displayName":"capability",
      "value":"monitoring"
    }
  },
  "tags": {
    "assoc_dest:tj_host" : ""
  }
]
}
```

3

Enable Monitoring for Previously Discovered Entities

Entities can be added from other Oracle Management Cloud services. The following topics cover how to allow Infrastructure Monitoring to monitor entities added via Log Analytics and Application Performance Management.

- [Auto-Discovery of Previously Discovered Entities](#)
- [Manual Discovery of Log Analytics Entities](#)

Auto-Discovery of Previously Discovered Entities

What is Auto-Discovery

Auto-Discovery allows you to enable Infrastructure Monitoring on entities that were previously added by non-monitoring services, such as Application Performance Management or Log Analytics, without further intervention. Enabling auto-discovery is carried out in two phases:

1. Prepare a Cloud agent for auto-discovery.
2. Register the entity/agent auto-discovery configuration from the Oracle Management Cloud console.

Prerequisites

- Oracle Management Cloud agents version 1.23 and greater are auto-discovery capable.
- If Cloud agents are upgraded, auto-discovery can start only after configuration metric collections from the selected Cloud agents have begun. This can take up to 24 hours depending on configuration metric collection schedule.
- Only Tomcat and WebLogic J2EE Server entity types can be auto-discovered for the current release. For Tomcat and WebLogic Server monitoring prerequisites, see [Prerequisites and Monitoring Credentials](#).
- If the WebLogic J2EE Server does not use the t3/t3s security protocol, then two auto-discovery attempts are made: One with "t3" and the second with "t3s"
- For targets that require secured communications, one of the following will be true:
 - Either all targets of the same entity type will be configured to use the same security settings for auto-discovery or,
 - all targets of the same entity type with the same "tag" will be configured to use the same security settings for auto-discovery. See [Prerequisites and Monitoring Credentials](#) for more information regarding monitoring credentials.

Setting Up Auto-Discovery

Preparing the Cloud Agent

The first step is to select and configure one or more agents for Auto-Discovery. This involves creating named global credentials for each Auto-Discovery supported entity type that requires credentials and then preparing the agent for target communication

1. Prepare the credential store:

a. Stop the agent

```
./omcli stop agent
```

b. Create the credential store:

```
./omcli add_credential_store agent -no_password
```

c. Restart the agent:

```
./omcli start agent
```

2. Prepare the JSON files for creating global named credentials. The following examples show JSON files for both Tomcat and WebLogic entity types.

Example: Tomcat

```
[{
  "name": "tomcat_creds",
  "type": "TomcatCreds",
  "globalName": "omc_tomcat.Credential",
  "description": "Dummy Credential for accessing Credential-
less Tomcat Targets used for Auto-Discovery",
  "properties": [{
    "name": "jmx_username",
    "value": "CLEAR[none]"
  },
  {
    "name": "jmx_password",
    "value": "CLEAR[none]"
  }
]
}]
```

Example: WebLogic

```
[{
  "name": "DomainCredsMonitoring",
  "type": "MonitorCreds",
  "globalName": "omc_weblogic_domain.Credential",
  "description": "Credential for accessing WebLogic Domain
Targets for Auto-Discovery",
  "properties": [{
    "name": "user_name",
    "value": "CLEAR[weblogic]"
  },
  {
    "name": "password",
    "value": "CLEAR[user_pw]"
  }
]
```

```
    ]
  ]]
```

3. Run OMCLI commands to create the named credentials.

Example: Tomcat

```
./omcli add_credentials agent -credential_file /myjsons/
tomcat_named_creds.json -allow_entityless
```

Example: WebLogic

```
./omcli add_credentials agent -credential_file /myjsons/
wls_autodiscovery_creds.json -allow_entityless
```

4. You can verify that the named credentials have been created by listing them.

```
./omcli list_credentials agent -global -usage MONITORING
```

5. Enable the named credentials.

Example: Tomcat

```
./omcli enable_credential agent omc_tomcat.Credential -global
```

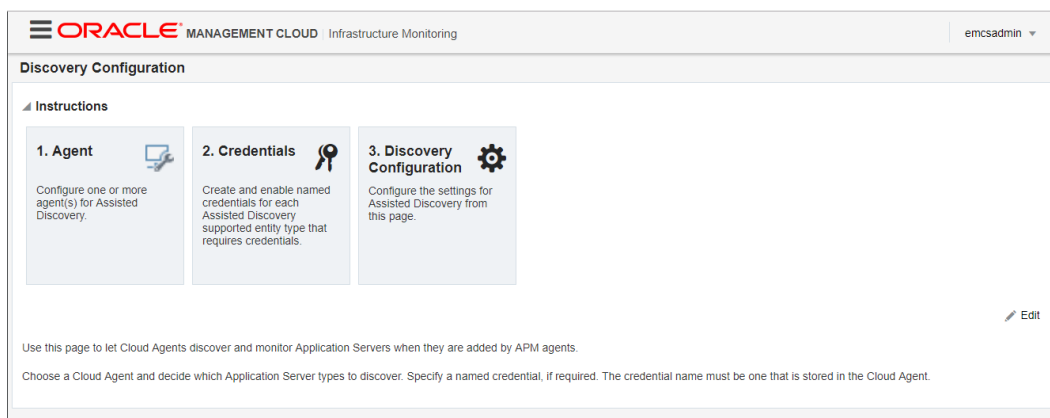
Example: WebLogic

```
./omcli enable_credential agent omc_weblogic_domain.Credential -global
```

Registering the Auto-Discovery Configuration from the Oracle Management Cloud Console

Once your selected agents have been configured you are ready to enable auto-discover. This step is to register the compatible entity types from the Oracle Management Cloud console.




1. From the Oracle Management Cloud console main menu, select **Monitoring** and then **Discovery Configuration**. The Discovery Configuration page displays.



ORACLE MANAGEMENT CLOUD Infrastructure Monitoring emcsadmin

Discovery Configuration

Instructions

- 1. Agent**  Configure one or more agent(s) for Assisted Discovery.
- 2. Credentials**  Create and enable named credentials for each Assisted Discovery supported entity type that requires credentials.
- 3. Discovery Configuration**  Configure the settings for Assisted Discovery from this page.

[Edit](#)

Use this page to let Cloud Agents discover and monitor Application Servers when they are added by APM agents.

Choose a Cloud Agent and decide which Application Server types to discover. Specify a named credential, if required. The credential name must be one that is stored in the Cloud Agent.

2. Click **Edit**. The Edit Discovery Configuration page displays.

The screenshot shows the 'Edit Discovery Configuration' page in Oracle Management Cloud. At the top, it says 'ORACLE MANAGEMENT CLOUD Infrastructure Monitoring' and 'emcsadmin'. Below that is the title 'Edit Discovery Configuration' and 'Save' and 'Cancel' buttons. A paragraph explains the purpose: 'Use this page to let Cloud Agents discover and monitor Application Servers when they are added by APIM agents. Choose a Cloud Agent and decide which Application Server types to discover. Specify a named credential, if required. The credential name must be one that is stored in the Cloud Agent.' Below this is a 'Cloud Agent' dropdown menu with the text 'Select a Cloud Agent'. Underneath is a table with two columns: 'Types to Discover' and 'Credentials'. The table has two rows: one for 'Tomcat' with 'Credentials, Optional' and one for 'WebLogic Server' with 'Credentials, Required'. Each row has a checkbox in the 'Types to Discover' column.

3. Choose a compatible agent from the drop-down list.

 **Note:**

Agents must be version 1.23 or greater.

4. Choose the entity types you want auto-discovered.
5. Register the named credentials. **IMPORTANT:** Credentials must match the corresponding *Named Credentials* created on the selected agent.
6. Click **Save** to save your changes.

Once you have enabled auto-discovery for the specific agent and entity type, whenever you add entities from other services, such as Application Performance Management or Log Analytics, those entities will be automatically discovered and monitored by Infrastructure Monitoring.

Manual Discovery of Log Analytics Entities

When monitoring entities using both Oracle Infrastructure Monitoring and Oracle Log Analytics, it's important to keep in mind which service you used first to discover the entities. If you discovered entities using Oracle Infrastructure Monitoring first, the Oracle Log Analytics entities are automatically discovered. However, if you discovered entities using Oracle Log Analytics first, you will need to edit the entity JSON files in order for Oracle Infrastructure Monitoring to discover these entities. The following examples illustrate how to update the JSON files.

Monitoring Oracle Databases with Oracle Infrastructure Monitoring and Oracle Log Analytics

Enabling infrastructure monitoring for an Oracle Database entity varies depending on the sequence you want to use Oracle Log Analytics and Oracle Infrastructure Monitoring. The following scenarios illustrate the most common implementations. For both scenarios, no Oracle Database entities were added to either Oracle Log Analytics or Oracle Infrastructure Monitoring. The Cloud agent resides locally on the database host.

Example 1: Enabling Oracle Infrastructure Monitoring and Oracle Log Analytics Concurrently

What you want: You want to use both Oracle Infrastructure Monitoring and Oracle Log Analytics to monitor an Oracle Database entity.

How to do it: Add the Oracle Database entity using the Oracle Infrastructure Monitoring JSON files:

- omc_oracle_db_sample.json
- omc_oracle_db_sample_creds.json

Once the Oracle Database entity has been added, the database will be monitored by Oracle Infrastructure Monitoring and a database instance will have been added for Oracle Log Analytics.

Example 2: Enabling Oracle Log Analytics First and Adding Oracle Infrastructure Monitoring Later

What you want: You only want to use Oracle Log Analytics for a newly added Oracle Database entity at this time. However, at some point in the future, you may want to monitor that same Oracle Database entity using Oracle Infrastructure Monitoring.

How to do it: Add the Oracle Database entity using the special JSON file *DB_Discovery_Credless.json* (supplied in the *omc_lasrv_entity_json_samples.zip*) that is set up to give you the option of later enabling that same Oracle Database for Oracle Infrastructure Monitoring.

 **Note:**

DB_Discovery_Credless.json file is a Oracle Log Analytics-specific JSON file that allows you to add Oracle Infrastructure Monitoring capability at a later time. It **cannot** be used interchangeably with Oracle Infrastructure Monitoring JSON files used to add Oracle database entities (shown in Example 1).

Example 3: Adding Oracle Infrastructure Monitoring

What you want: You now want to enable Oracle Infrastructure Monitoring for the Oracle Database entity you previously added to Oracle Log Analytics.

How to do it: Modify a copy of *DB_Discovery_Credless.json* file, and update the Oracle Database entity added in Example 2 using the following steps:

1. Edit your copy of the *DB_Discovery_Credless.json* file to include only the following information. For this example, we'll call the file *omc_oracle_yourdb_credless.json*.
 - a. Entity name: Use the same name used for Oracle Log Analytics.
 - b. Entity type: This is a reserved field name, unique for each entity type. Use the same type used for Oracle Log Analytics.

- c. Under attributes, add the `credentialRefs`, marked as `credential_refs_id`, as follows. Note: `"credential_refs_id"` must match credential ID in credential file.

```
"credentialRefs":[
    "credential_refs_id"
]
```

- d. Under property, keep only the `capability` section, marked as `monitoring`,

```
"capability":{
    "displayName":"capability",
    "value":"monitoring"
}
```

2. Edit the credentials JSON files, as documented for monitoring. For this example, let's call the file `omc_oracle_yourdb_creds.json`.

```
"credentials" : [
    {
        "id" : "SQLCreds",
        "name" : "SQLCreds",
        "credType" : "DBCreds",
        "properties" : [
            {
                "name" : "DBUserName",
                "value" : "CLEAR[Your Database User Name]"
            },
            {
                "name" : "DBPassword",
                "value" : "CLEAR[Your Database Password]"
            },
            {
                "name" : "DBRole",
                "value" : "CLEAR[Normal]"
            }
        ]
    }
]
```

For target communications in secured mode, you need additional entity-type specific information, so make sure you select the appropriate credentials sample JSON file to edit.

3. Update the entity with the edited JSON files:

```
omcli update_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE]
```


For example, on a UNIX system, from the <AGENT_BASE_DIR>/agent_inst/bin run:

```
./omcli update_entity agent omc_oracle_yourdb_credless.json -  
credential_file omc_oracle_yourdb_creds.json
```

The following complete JSON example shows the *omc_oracle_mydb_credless.json* file with the required updates to enable Oracle Infrastructure Monitoring for an Oracle Database entity.

```
{  
  "entities": [{  
    "name": "DB_Name",  
    "type": "omc_oracle_db",  
    "displayName": "DB_Name",  
    "timezoneRegion": "PST",  
    "credentialRefs": [  
      "SQLCreds"  
    ],  
    "properties": {  
      "host_name": {  
        "displayName": "Host Name",  
        "value": "host.example.com"  
      },  
  
      "port": {  
        "displayName": "Port",  
        "value": "1521"  
      },  
  
      "sid": {  
        "displayName": "SID",  
        "value": "sid_name"  
      },  
  
      "protocol": {  
        "displayName": "Protocol",  
        "value": "tcp"  
      },  
  
      "is_cluster": {  
        "displayName": "Is Cluster",  
        "value": "FALSE"  
      },  
  
      "service_name": {  
        "displayName": "Service Name",  
        "value": "host1.example.com"  
      },  
  
      "global_name": {  
        "displayName": "Global Name",  
        "value": "host1.example.com"  
      },  
      "capability": {  
        "displayName": "capability",
```

```

        "value": "monitoring"
    }
},

"associations": [{
    "assocType": "omc_contains",
    "sourceEntityName": "DB_Name",
    "sourceEntityType": "omc_oracle_db",
    "destEntityName": "DB_Name/sid_name",
    "destEntityType": "omc_oracle_db_instance"
}]
},

{
    "name": "DB_Name/sid_name",
    "type": "omc_oracle_db_instance",
    "displayName": "DB_Name/sid_name",
    "timezoneRegion": "PST",
    "properties": {
        "host_name": {
            "displayName": "Host Name",
            "value": "host.example.com"
        },

        "audit_dest": {
            "displayName": "Audit Dest",
            "value": "/opt/ORCLemaas/InfrastructureSoftware/oracledb/base/
admin/orcl12c/adump"
        },

        "instance_name": {
            "displayName": "Instance Name",
            "value": "sid_name"
        },

        "adr_home": {
            "displayName": "ADR Home",
            "value": "/opt/ORCLemaas/InfrastructureSoftware/oracledb/base"
        }
    }
}
]
}

```

Adding a WebLogic Server Entity to Oracle Log Analytics and Later Oracle Infrastructure Monitoring

The process of adding a WebLogic Server entity to Oracle Log Analytics and later adding it to Oracle Infrastructure Monitoring is shown in the Oracle by Example tutorial, [“Adding a WebLogic Server Entity to Oracle Log Analytics and Later Oracle Infrastructure Monitoring”](#).

The process shown in this tutorial also demonstrates the general process used to add Oracle Infrastructure Monitoring to an existing Oracle Log Analytics deployment.

4

Extend Monitoring Capability with Metric Collectors

You can extend Oracle Infrastructure Monitoring capability by using open source metric collector agents to collect additional types of metric data.

Oracle Management Cloud supports two collector agents: `collectd` and Telegraf.

- [Expand Monitoring with collectd](#)
- [Expand Monitoring with Telegraf](#)

Expand Monitoring with `collectd`

`collectd` is an open source tool used to collect metric data from various sources (such as operating systems, applications, external devices, and log files).

When installed, `collectd` expands the scope of Oracle Infrastructure Monitoring by increasing the number and type of metrics that can be collected via `collectd`'s large and ever increasing open source plugin library.

Enabling `collectd` for Oracle Management Cloud, involves configuring `collectd` to send metric data to the Cloud Agent. `collectd` metrics can be automatically mapped to Oracle Management Cloud metrics. When `collectd` auto-mapping has been performed, a new entity type is created for each plugin.

When viewing these auto-created entities from the Oracle Management Cloud console, you can determine whether Oracle Management Cloud is receiving the `collectd` metrics through the following entity availability states:

- UP: Oracle Management Cloud is receiving metrics from `collectd`.
- DOWN: Oracle Management Cloud is not receiving metrics from `collectd`.

You can modify these auto-created entity types via REST API.



Note:

REST API documentation access is currently limited to approved customers. Contact your Oracle Support or Sales Representative for more information about accessing and using REST API documentation.

Auto-created entities require a Standard Edition/Enterprise Edition license. For more information, see [Enabling License Editions for Oracle Management Cloud and Oracle Management Cloud Offerings](#).

The following steps illustrate how to use `collectd` with Oracle Management Cloud.

Important: All tasks shown within step examples are required unless otherwise noted.

Step	Illustrative Example
1. Configure READ plugins to read data into collectd.	Example Read Plugin — Processes Plugin
2. Integrate collectd with the Cloud agent. Here, you configure collectd to send metrics to the Cloud agent receiver over HTTPS and the permit the Cloud agent to monitor it using <code>collectdctl</code> .	Example: Configuring collectd
3. Add a generic metric collector entity to Oracle Management Cloud.	Example: Generic Metric Collector Entity Type (Auto-mapping)

Example Read Plugin — Processes Plugin

The [processes plugin](#) collects the number of processes, grouped by their state (e. g. running, sleeping, zombies, etc.). Additionally, it can collect detailed statistics about selected processes, grouped by name. For more information about specifying the *selected processes*, refer to the `collectd.conf(5)` man page. For example, the following configuration snippet can be used to collect detailed statistics on Apache HTTP server processes, the Cloud agent processes, and collectd's own process running on a particular system.

Processes Plugin Configuration Example

```
LoadPlugin processes
<Plugin processes>
    Process "httpd"
    ProcessMatch "omc-cloud-agent"
"java .*oracle.sysman.gcagent.tmmain.TMMain"
    Process "collectd"
</Plugin>
```

Example: Configuring collectd

In this example, we'll configure collectd to push metrics to the cloud agent receiver via collectd's `write_http` plugin.



Note:

The examples shown in this section are only applicable to collectd version 5.5 and above.

The generic metric collector's receiver listens to requests sent to `https://<host>:<port>/emd/receiver/gmc`. Here, `<port>` is same port shown in the URL returned by running the `omcli status agent` command on the cloud agent host.

On the collectd side, the same information is provided in its configuration file, as shown in the following example.

write_http Plugin Configuration

```
<Plugin write_http>
  <Node "omc">
    URL "https://127.0.0.1:1899/emd/receiver/gmc"
    Format "JSON"
    StoreRates true
    VerifyPeer false
  </Node>
</Plugin>
```

By setting *StoreRates* to *true*, collectd is configured to send rates rather than counter values to Oracle Management Cloud. The *POST* method is used to send and receive metrics.

The *collectdctl* control interface communicates with the collectd agent process using UNIX domain sockets. To have the cloud agent monitor collectd, the *unixsock* plugin is enabled and configured as shown in *unixsock Plugin Configuration*.



Note:

If collectd is remotely located relative to the cloud agent host, specify the cloud agent host name instead of the loop-back address of 127.0.0.1.

unixsock Plugin Configuration

```
<Plugin unixsock>
  SocketFile "/opt/collectd/var/run/collectd-unixsock"
  SocketGroup "collectd"
  SocketPerms "0660"
  DeleteSocket false
</Plugin>
```



Note:

unixsock plugin configuration (along with adding an agent user to the SocketGroup) is only required if collectd is co-located with cloud agent and should be monitored by it.

As shown in the configuration above, UNIX domain sockets use file-based permissions, unlike internet sockets. In case the cloud agent user is not the same as the user running the collectd process, you need to ensure that the cloud agent user has both read and write (06) permission on the socket. One way to ensure this is by adding the cloud agent user to *SocketGroup* ("collectd" in the above example).

The cloud agent user is the user who started the cloud agent. You can determine who the cloud agent user is by running the following command:

```
omcli status agent
```

If the group specified for `SocketGroup` does not exist on the host, you'll need to add one as shown below.

```
$ sudo /usr/sbin/groupadd collectd
```

Add a Cloud Agent User to a SocketGroup

In the following example, you are adding a cloud agent to `SocketGroup` in `collectd`.

```
$ sudo /usr/sbin/usermod -a -G collectd <cloud agent user name>
```

With group permissions in effect, the cloud agent user will be able to run `collectdctl` commands that communicate with `collectd` agent.

Additionally, to protect the receiver URI, a new credential with username and password can be added to the cloud agent using the `omcli` command line interface.

Add HTTP Basic Authentication Credentials (Optional)

```
$ omcli add_credentials agent -credential_file http_receiver_auth.json
```

Example content for the `http_receiver_auth.json` is shown below.

```
[
  {
    "entity": "Lama.mycompany.com:1899",
    "name": "omc-receiver-cred",
    "type": "HttpReceiver-Auth",
    "globalName": "HttpReceiver.basic",
    "description": "Internal authorization for the HTTP receiver",
    "properties": [
      { "name": "username", "value": "CLEAR[scott]" },
      { "name": "password", "value": "CLEAR[tiger]" } ]
  }
]
```

Here, the cloud agent entity is specified in "`<type>.<name>`" format (using a period (.) as a separator) where the `<name>` includes the agent's listening port. The properties `username` and `password` are set to `scott` and `tiger` respectively. The sender will be required to authenticate with these credentials.

The same username and password are then made known to `collectd`'s `write_http` plugin by adding `User` and `Password` fields to its configuration as shown in the next section.

write_http Plugin Configuration for HTTP Basic Authentication (Optional)

```
<Plugin write_http>
  <Node "omc">
    URL "https://127.0.0.1:1899/emd/receiver/gmc"
    Format "JSON"
    StoreRates true
    VerifyPeer false
    User "scott"
```

```

        Password "tiger"
      </Node>
</Plugin>

```

Example: Generic Metric Collector Entity Type (Auto-mapping)

The following example shows how to define a generic metric collector entity representing a *collectd* agent that is locally monitored with metric auto-mapping functionality enabled:



Note:

The metric auto-mapping functionality is enabled in the following example by setting the *omc_auto_map* property to *TRUE*.

Example 4-1 Entity of Generic Metric Collector Type - Locally Monitored

```

{
  "entities":
  [
    {
      "name": "<Your name for the collectd collector>",
      "type": "omc_generic_metric_collector",
      "displayName": "<Your display name for the collectd collector>",
      "timezoneRegion": "<Your timezone>",
      "properties":
      {
        "host_name":
        {
          "displayName": "Host Name",
          "value": "<Your name of the host where collectd is installed>"
        },

        "omc_query_interface_path":
        {
          "displayName": "Query Interface Path",
          "value": "<Full path to the collectdctl file>"
        },

        "omc_filter_expression":
        {
          "displayName": "Filter Expression",
          "value": "{$.[?(@.host=='<Value of the host field in the metric
payload sent by collectd>')]}"

```

```
{
  "displayName": "capability",
  "value": "monitoring"
},

"omc_monitored":
{
  "displayName": "Cloud Agent Monitored",
  "value": "TRUE"
},

"omc_product_name":
{
  "displayName": "Product Name",
  "value": "collectd"
},

"omc_product_vendor":
{
  "displayName": "Product Vendor",
  "value": "Florian octo Forster, et al."
},

"omc_product_version_query_arg":
{
  "displayName": "Product Version Query Argument",
  "value": "-h"
},

"omc_product_version_regex":
{
  "displayName": "Product Version Regular Expression",
  "value": "^collectd (.+), http"
},

"omc_metrics_query_arg":
{
  "displayName": "Metrics Query Argument",
  "value": "listval"
},

"omc_response_query_arg":
{
  "displayName": "Response Query Argument",
  "value": "listval"
},

"omc_use_exit_code_for_response":
{
  "displayName": "Use exit code for response",
  "value": "TRUE"
},

"omc_protocol":
{
```



```
        "displayName": "Protocol",
        "value": "https"
    },

    "omc_payload_format":
    {
        "displayName": "Payload Format",
        "value": "json"
    },

    "omc_receiver_uri_path":
    {
        "displayName": "Receiver URI Path",
        "value": "/emd/receiver/gmc"
    }
}
]
}
```

For additional configurations, metric schema mapping, and troubleshooting, see [Additional collectd Configurations and Information](#).

Expand Monitoring with Telegraf

Telegraf allows you to add support for collecting metrics with minimal memory footprint.

Telegraf is a plugin-driven agent used to collect, process, aggregate, and output metric data. Using Telegraf expands the scope of Oracle Infrastructure Monitoring by increasing the number and type of metrics that can be collected via Telegraf's large and ever increasing plugin library.

Enabling Telegraf for Oracle Management Cloud involves configuring Telegraf to send metric data to the cloud agent. Telegraf metrics can be automatically mapped to Oracle Management Cloud metrics with a new entity type created for each input plugin.



Note:

Telegraf integration with requires Oracle Management Cloud agent version 1.40 or greater.

When viewing these auto-created entities from the Oracle Management Cloud console, you can determine whether Oracle Management Cloud is receiving the Telegraf metrics through the following entity availability states:

- *UP*: Oracle Management Cloud is receiving metrics from Telegraf.
- *DOWN*: Oracle Management Cloud is not receiving metrics from Telegraf.

You can modify these auto-created entity types via REST API.

**Note:**

REST API documentation access is currently limited to approved customers. Contact your Oracle Support or Sales Representative for more information about accessing and using REST API documentation.

Auto-created entities require a Standard Edition/Enterprise Edition license. For more information, see [Enabling License Editions for Oracle Management Cloud](#) and [Oracle Management Cloud Offerings](#).

The following steps illustrate how to use Telegraf with Oracle Management Cloud. All steps are required.

Step	Illustrative Example
1. Configure input plugins to read data into Telegraf.	Example Input Plugin: Processes Plugin
2. Configure Telegraf to send metrics to the cloud agent over HTTPS.	Example: Configure Telegraf for Oracle Management Cloud Integration
3. Add a generic metric collector entity representing the Telegraf agent to Oracle Management Cloud. This step is needed to map the availability status of Telegraf, its version, specify how the cloud agent should handle metrics sent by the Telegraf agent, etc.	Example Generic Metric Collector Entity (Telegraf)

If you don't see any metrics from Telegraf after performing the integration steps, see [Troubleshooting Telegraf Metric Collection](#).

Example Input Plugin: Processes Plugin

The Processes input plugin gathers information about the total number of processes and groups them by status (zombie, sleeping, running, etc.).

```
# Get the number of processes and group them by status
[[inputs.processes]]
  # no configuration
```

For more information about the processes input plug-in, see [Processes Input Plugin](#).

Example: Configure Telegraf for Oracle Management Cloud Integration

The following example demonstrates how to integrate Telegraf with Oracle Management Cloud.

The generic metric collector's receiver in the cloud agent listens to requests sent to `https://<host>:<port>/emd/receiver/gmc`. Here, `<port>` is same port shown in the URL returned by running the `omcli status agent` command on the cloud agent host.

On the Telegraf side, the same information is provided in its configuration file, as shown in the following example.

```
# Global tags can be specified here in key="value" format.
[global_tags]
  collector = "telegraf"

[[outputs.http]]
  url = "https://127.0.0.1:1899/emd/receiver/gmc"
  method = "POST"
  insecure_skip_verify = true
  data_format = "json"
```

The POST method is used to send and receive metrics. Also, the above configuration sets a global tag called `collector` to the value `telegraf`. This setting is essential and is used by cloud agent to recognize metrics sent by Telegraf.

Note:

If Telegraf is remotely located relative to the cloud agent host, specify the cloud agent host name instead of the loop-back address of 127.0.0.1. More on this configuration is covered in [Receive Metrics from a Remote Telegraf Collector](#).

Metrics sent by Telegraf are posted against entities in Oracle Management Cloud whose names are derived using the value of the host tag set in Telegraf's payload sent to cloud agent. This value is usually the short host name by default. If multiple hosts can exist with the same short host name (in different domains) within your tenancy, you also need to set the `hostname` tag to the host's FQDN to disambiguate the entity names that will be created in Oracle Management Cloud.

```
[agent] hostname="myhost.myco.com"
```

Incorporate HTTP Basic Authentication Scheme (Optional)

To protect the receiver URI, a new credential with username and password can be added to the cloud agent using the `omcli add_credentials` command:

```
omcli add_credentials agent -credential_file http_receiver_auth.json
```

Example content for the `http_receiver_auth.json` is shown below.

```
[
  { "entity": "Lama.mycompany.com:1899",
    "name": "omc-receiver-cred",
    "type": "HttpReceiver-Auth",
    "globalName": "HttpReceiver.basic",
    "description": "Internal authorization for the HTTP receiver",
    "properties": [ { "name": "username", "value": "CLEAR[scott]" },
                   { "name": "password", "value": "CLEAR[tiger]" } ]
  }
]
```

Here, the cloud agent entity is specified in `<type>.<name>` format (using a period (.) as a separator) where the `<name>` includes the agent's listening port. The properties `username` and `password` are set to `scott` and `tiger` respectively. The sender will be required to authenticate with these credentials.

Share this *secret* with Telegraf by adding the `username` and `password` parameters to the `outputs.http` section in the `telegraf.conf` file as shown below.

```
[[outputs.http]]
  url = "https://<Cloud Agent's Host Name>:<Cloud Agent's HTTP SSL
Port>/emd/receiver/gmc"
  method = "POST"
  insecure_skip_verify = true
  data_format = "json"
  username = "scott"
  password = "tiger"
```

In order for a cloud agent to be able to monitor Telegraf, Telegraf must be configured to run as a service.

Example Generic Metric Collector Entity (Telegraf)

The following examples show how to define a generic metric collector entity representing a Telegraf agent that is locally monitored by a cloud agent on Unix and Windows.

Example: Unix and Unix-variants

```
{
  "entities":
  [
    {
      "name": "<Your name for the Telegraf collector>",
      "type": "omc_generic_metric_collector",
      "displayName": "<Your display name for the Telegraf collector>",
      "timezoneRegion": "<Your timezone>",
      "properties":
      {
        "host_name":
        {
          "displayName": "Host Name",
          "value": "<Your name of the host where Telegraf is
installed>"
        },

        "omc_query_interface_path":
        {
          "displayName": "Query Interface Path",
          "value": "<Full path to the telegraf file. Eg. /usr/bin/
telegraf>"
        },

        "omc_filter_expression":
        {
```

```
        "displayName": "Filter Expression",
        "value": "{$.[?(@.host=='<Value of the host tag in the metric
payload sent by Telegraf>')]}"
    },

    "omc_auto_map":
    {
        "displayName": "Automatically Map Metrics",
        "value": "TRUE"
    },

    "capability":
    {
        "displayName": "capability",
        "value": "monitoring"
    },

    "omc_monitored":
    {
        "displayName": "Cloud Agent Monitored",
        "value": "TRUE"
    },

    "omc_product_name":
    {
        "displayName": "Product Name",
        "value": "telegraf"
    },

    "omc_product_vendor":
    {
        "displayName": "Product Vendor",
        "value": "InfluxData Inc."
    },

    "omc_product_version_query_arg":
    {
        "displayName": "Product Version Query Argument",
        "value": "--version"
    },

    "omc_product_version_regex":
    {
        "displayName": "Product Version Regular Expression",
        "value": "^Telegraf (.+) \\\"("

    "omc_metrics_query_arg":
    {
        "displayName": "Metrics Query Argument",
        "value": "--test"
    },

    "omc_response_query_command":
    {
```

```

        "displayName": "Response Query Command",
        "value": "/sbin/service telegraf status"
    },

    "omc_use_exit_code_for_response":
    {
        "displayName": "Use exit code for response",
        "value": "TRUE"
    },

    "omc_protocol":
    {
        "displayName": "Protocol",
        "value": "https"
    },

    "omc_payload_format":
    {
        "displayName": "Payload Format",
        "value": "json"
    },

    "omc_receiver_uri_path":
    {
        "displayName": "Receiver URI Path",
        "value": "/emd/receiver/gmc"
    }
}
]
}

```

Example: Microsoft Windows

```

{
  "entities":
  [
    {
      "name": "<Your name for the Telegraf collector>",
      "type": "omc_generic_metric_collector",
      "displayName": "<Your display name for the Telegraf collector>",
      "timezoneRegion": "<Your timezone>",
      "properties":
      {
        "host_name":
        {
          "displayName": "Host Name",
          "value": "<Your name of the host where Telegraf is
installed>"
        },

        "omc_query_interface_path":
        {
          "displayName": "Query Interface Path",

```

```
        "value": "<Full path to the telegraf.exe file escaping backslash
characters. Eg., C:\\Program Files\\Telegraf\\telegraf.exe>"
    },

    "omc_filter_expression":
    {
        "displayName": "Filter Expression",
        "value": "{$.[?(@.host=='<Value of the host tag in the metric
payload sent by Telegraf>')]}"
```

```

        "displayName": "Metrics Query Argument",
        "value": "--console --test"
    },

    "omc_response_query_command":
    {
        "displayName": "Response Query Command",
        "value": "sc query telegraf"
    },

    "omc_response_regex":
    {
        "displayName": "Response Regular Expression",
        "value": "STATE.*RUNNING"
    },

    "omc_protocol":
    {
        "displayName": "Protocol",
        "value": "https"
    },

    "omc_payload_format":
    {
        "displayName": "Payload Format",
        "value": "json"
    },

    "omc_receiver_uri_path":
    {
        "displayName": "Receiver URI Path",
        "value": "/emd/receiver/gmc"
    }
    }
}
]
}

```



Note:

The metric auto-mapping functionality is enabled in this example by setting the `omc_auto_map` property to `TRUE`. **Manual mapping of Telegraf metrics is not currently supported.**

Use the `omcli add_entity` command to add the entity to the cloud agent.

```
omcli add_entity agent <entityDefinitionJsonFilePath>
```


Troubleshooting Telegraf Metric Collection

If Telegraf metric data does not appear in Oracle Management Cloud as expected, follow the basic debugging procedure show below.

Troubleshooting Procedure

1. Ensure that the generic metric collector entity was added successfully to the cloud agent by the `omcli add_entity` command. If it is not showing up in the metric browser, run the `status_entity omcli` command as shown below:

```
omcli status_entity agent <entityDefinitionJsonFilePath>
```

Validation errors, if any, will be shown in the command output.

2. Enable trace level logging in `emd.properties`. Set the following two properties:

```
Logger._enableTrace=true  
Logger.sdklog.level=DEBUG
```

and bounce the cloud agent. Run the tail command on the `gcagent_sdk.trc` file in the agent's log directory.

3. From the log file you should see the complete payload received by agent from Telegraf, which metrics are in turn being sent by receiver to Oracle Management Cloud, and which metrics are unmapped.
Search for "gmcReceiver received payload" in the log file to see the full payload received. If this line is not seen in the log file, the agent may not be receiving data from Telegraf. If this is the case:

- Check if Telegraf is running.
- Check that the intended input plugins are enabled and Telegraf is able to collect their metrics by running the `telegraf --test` command as shown in the following example.

```
$ telegraf --test  
2019/03/04 21:00:09 I! Using config file: /etc/telegraf/telegraf.conf  
> cpu,collector=telegraf,cpu=cpu0,host=myhost.myco.com  
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_i  
rq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_u  
ser=0 1551762010000000000  
> cpu,collector=telegraf,cpu=cpu1,host=myhost.myco.com  
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_i  
rq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_u  
ser=0 1551762010000000000  
> cpu,collector=telegraf,cpu=cpu2,host=myhost.myco.com  
usage_guest=0,usage_guest_nice=0,usage_idle=98.00000004470348,usage_io  
wait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_sy  
stem=0,usage_user=1.99999998952262 1551762010000000000  
> cpu,collector=telegraf,cpu=cpu3,host=myhost.myco.com  
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_i  
rq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_u  
ser=0 1551762010000000000  
> cpu,collector=telegraf,cpu=cpu-total,host=myhost.myco.com  
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_i
```

```

rq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_user=0 1551762010000000000
> mem,collector=telegraf,host=myhost.myco.com
active=6735482880i,available=11130187776i,available_percent=73.67
584678266645,buffered=3569352704i,cached=7279378432i,commit_limit
=22233530368i,committed_as=4000460800i,dirty=839680i,free=2814566
40i,high_free=0i,high_total=0i,huge_page_size=2097152i,huge_pages
_free=0i,huge_pages_total=0i,inactive=5336559616i,low_free=0i,low
_total=0i,mapped=1415385088i,page_tables=116322304i,shared=134002
6880i,slab=2446262272i,swap_cached=14417920i,swap_free=1436728524
8i,swap_total=14680047616i,total=15106969600i,used=3976781824i,us
ed_percent=26.324153217333542,vmalloc_chunk=35184301154304i,vmall
oc_total=35184372087808i,vmalloc_used=50819072i,wired=0i,write_ba
ck=0i,write_back_tmp=0i 1551762010000000000

```

If the test is successful, but metrics are still not being received by cloud agent, check that the HTTP output plugin has been configured correctly. Check the host and port in the URL. Check Telegraf's log file or syslog for errors if any reported from outputs.http. Check if other software applications such as SELinux, anti-virus, or a firewall may be blocking Telegraf's ability to write metrics to the cloud agent's port.

4. Search for payload level summary lines in the log file which starts with the "Source Metrics" line. These lines should give a summary count of statistics such as how many metrics are being received in each payload, how many have been sent to Oracle Management Cloud, or how many are unmapped.

Payload Level Summary Logging Example - gcagent_sdk.trc

Log of Payload Level Summary

```

2019-03-04 21:45:04,613 [401336:9A108C02] DEBUG - Source Metrics: 18
2019-03-04 21:45:04,613 [401336:9A108C02] DEBUG -
SEND_METRIC_GROUP_CALLED: 18

```

If the summary shows **SEND_METRIC_GROUP_CALLED: <count>**, that's normal.

If the summary shows

NO_ASSOC_GMC_ENTITY_WITH_MONITORING_CAPABILITY: <count>, then check that `omc_filter_expression` of the generic metric collector (gmc) entity allows the payload to filter through. Ensure that the name of the host field (if any) specified in the `omc_filter_expression` property exactly matches the host field's value in the payload. Also ensure that the gmc entity has either standard or enterprise license. License can be checked from Oracle Management Cloud's Administration UI.

If the summary shows **METRIC_UPLOAD_RATE_LIMIT_EXCEEDED: <count>**, then <count> metrics in the payload were down-sampled. They were not sent to up Oracle Management Cloud. This is expected if the sending interval is anything lower than once a minute (interval "60s" in the telegraf.conf file).

If the summary shows **WAITING_FOR_MAPPING_METADATA: <count>**, then <count> metrics in the payload are waiting for auto-map processing to complete. This is a transient state only expected in the automatic mapping case. Auto-map processing can take a few minutes to a tens of minutes to complete.

If the summary shows **SKIPPED_DUPLICATE_METRIC_POST**: <count>, then <count> metrics in the payload were skipped because multiple metric records were detected for any given entity, metric group and timestamp. In some cases this may be OK, such as when the payload contains redundant records which were skipped. In other cases, this may require user to tweak which tag(s) are used for entity identification by manually specifying `entity_identifier` in telegraf configuration file. In other cases, this may require tweaking the input plugin configuration or may even be a mapping limitation. For eg., ensure that the process name or pattern specified for procstat plugin captures a single process (PID). Ingesting procstat metrics for multiple processes in Oracle Managed Cloud is currently not supported and will result in skipped posts as seen in the log file.

If the summary shows **SKIPPED_AGGREGATE_METRIC_POST**: <count>, then <count> metrics were skipped because they are aggregates. Ingestion of aggregate metrics such as sum, min, max, mean, count, histograms, etc. from Telegraf is currently not supported.

5. If **SEND_METRIC_GROUP_CALLED**: <count> is seen, you should eventually start seeing entities on the monitoring service UI with type same as the Telegraf plugin name and entity name containing the Telegraf host's name (as obtained from the host field within the payload sent by Telegraf to Cloud agent). If you do not see such an entity, it's possible that the entity has been created, but has not been granted Standard or Enterprise license. This can be fixed by adding a license from the License Administration UI. From the Oracle Management Cloud console, select the Administration > Entities Configuration > Licensing link. From this page, look at the **Unlicensed Entities** link. If it shows the auto-created entity, assign License Edition = Standard or Enterprise and click Save. To ensure this happens automatically in future, set the **License Auto-Assignment** to *Standard or Enterprise*.
6. Once the auto-created entity shows up on the list of entities in the monitoring service UI, drill down into the entity to see the auto-mapped metrics. Only the availability metric will be shown by default. On the **Performance Charts** tab, Click **Options** > **Choose Metrics** to select the auto-created metrics for viewing their charts. Metric alert rules based on availability and threshold can also be defined on these performance metrics and are expected to work similar to alerts on metrics natively collected by Oracle Cloud agent. Anomaly alerts is disabled out of the box for Telegraf metrics auto-mapped in Oracle Manged Cloud
7. When debugging is no longer required, turn off trace level logging and set the SDK log level to INFO. Set the following in `emd.properties`.

Reset Log Level

```
Logger._enableTrace=false  
Logger.sdklog.level=INFO
```

5

Expand Monitoring Capability with Custom Metrics

Custom metrics allow you to create full-fledged metrics on any entity type that is monitored by a cloud agent. Custom metrics let you extend Oracle Management Cloud monitoring capabilities to monitor conditions specific to your IT environment. This provides you with a comprehensive view of your environment.

Creating custom metrics lets you simplify your IT organization's operational processes by leveraging Oracle Infrastructure Monitoring as the single central monitoring tool for your entire cloud environment instead of relying on other monitoring tools to provide this supplementary monitoring.

For instructions on obtaining and using REST APIs for Oracle Management Cloud Agent-based Custom Metrics, see [Oracle Management Cloud: Use Agent-Based Custom Metrics REST APIs \(Doc ID 2723626.1\)](#).

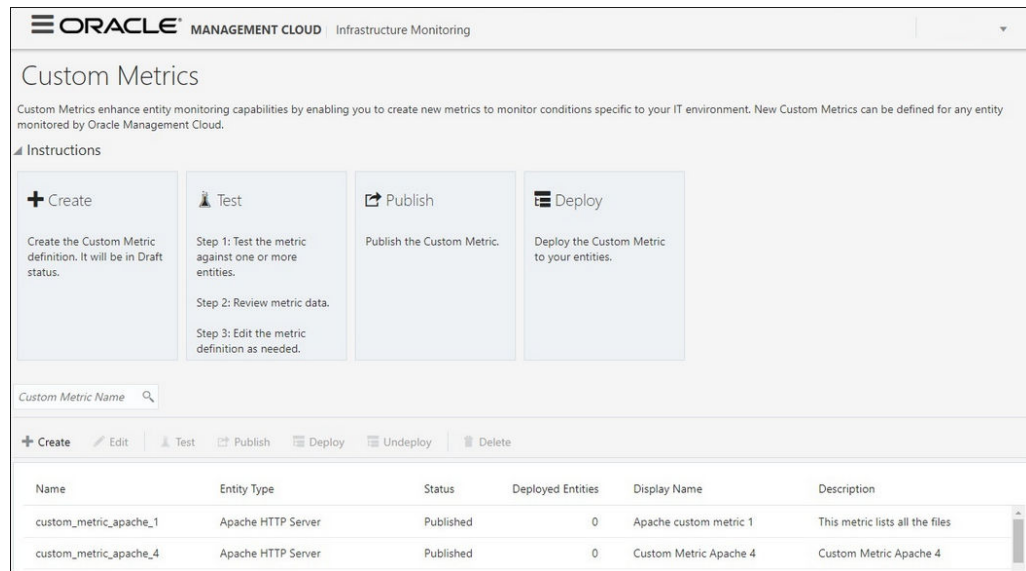


Note:

Custom metrics are not applicable to entities harvested from Enterprise Manager Cloud Control by the data collector.

Important: You can only create custom metrics for entity types that are monitored by cloud agents version 1.32 and greater.

You manage custom metrics from the Custom Metrics page. This page lists all existing custom metrics in addition to allowing you to create, edit, test, publish, and deploy custom metrics.

Figure 5-1 Custom Metrics

The screenshot displays the Oracle Management Cloud Infrastructure Monitoring interface for Custom Metrics. It includes a header with the Oracle logo and 'MANAGEMENT CLOUD Infrastructure Monitoring'. The main heading is 'Custom Metrics', followed by a brief description: 'Custom Metrics enhance entity monitoring capabilities by enabling you to create new metrics to monitor conditions specific to your IT environment. New Custom Metrics can be defined for any entity monitored by Oracle Management Cloud.' Below this is an 'Instructions' section with four steps: 'Create' (Create the Custom Metric definition. It will be in Draft status.), 'Test' (Step 1: Test the metric against one or more entities. Step 2: Review metric data. Step 3: Edit the metric definition as needed.), 'Publish' (Publish the Custom Metric.), and 'Deploy' (Deploy the Custom Metric to your entities.). A search bar for 'Custom Metric Name' is present. Below the instructions is a toolbar with buttons for '+ Create', 'Edit', 'Test', 'Publish', 'Deploy', 'Undeploy', and 'Delete'. At the bottom is a table with the following data:

Name	Entity Type	Status	Deployed Entities	Display Name	Description
custom_metric_apache_1	Apache HTTP Server	Published	0	Apache custom metric 1	This metric lists all the files
custom_metric_apache_4	Apache HTTP Server	Published	0	Custom Metric Apache 4	Custom Metric Apache 4

The cornerstone of the custom metric is the *adapter*. Adapters provide a means to gather data about agent-monitored entities using specific protocols. Adapter availability depends on the entity type your custom metric monitors. For a list of available adapters and adapter-related reference information, see [Custom Metric Collection Methods and Metric Columns](#).

Custom Metric Lifecycle

Developing a custom metric follows the same phases you would expect from any programmatic customization.

Creating a custom metric involves the following phases:

- Develop your custom metric.
- Test your custom metric.
- Publish and deploy your custom metric.

Develop Your Custom Metric

The first step is to define your monitoring requirements. This includes deciding the entity type, what data needs to be collected, and what mechanism (adapter) can be used to collect that data. After making these decisions, you are ready to begin developing your custom metric. Oracle Management Cloud provides an intuitive user interface to guide you through the creation process.

ORACLE MANAGEMENT CLOUD | Infrastructure Monitoring

Create Custom Metric

Submit Cancel

Basic Properties Adapter Properties Metric Columns Review

Basic Properties

General Properties

* Metric Name
Custom Metric Name can only contain alpha-numeric characters and underscore.

* Display Name

Description

Metric Type: Performance

* Entity Type: *Select an Entity Type*
Entity Type for which the Custom Metric is created

* Adapter

Collection Schedule

* Collection Frequency: 15 Minutes

When you have completed working on your custom metric, you can click **Submit**. The newly created custom metric appears in the list of custom metrics on the Custom Metrics main page where it can be accessed for further editing.



Note:

You can only edit a custom metric while it is in draft status.

The custom metric UI allows you to develop and refine your custom metric in a completely editable format. And more importantly, allows you to interactively test your metric against selected targets without having first to deploy the custom metric to a dedicated test environment. The **Test** page allows you to run real-time metric evaluations to ensure there are no syntactical errors in your script or custom metric definition.

Test Your Custom Metric

Once your custom metric returns the expected data during real-time target testing, you are ready to test its robustness and actual behavior in Oracle Management Cloud by deploying it against entities and start collecting data. This step involves selecting your editable custom metric shown in the Custom Metrics main page and running the metric against a valid entity from the Test page.

Publish and Deploy Your Custom Metric

After rigorous testing, your custom metric is ready for deployment to your production environment. Until this point, your custom metric is only viewable by you, the metric creator. To make it accessible to all Oracle Management Cloud administrators, it must be published. Simply select the desired custom metric from list on the Custom Metrics page and click **Publish**.

Now that your custom metric has been made public, it can be deployed to intended entities in your production environment.

Working with Custom Metrics

Most all custom metric operations can be carried out from the Custom Metrics UI.

Important: You must have Oracle Management Cloud Administrator privileges in order to create, edit, view, test, delete, publish, deploy or undeploy custom metrics.

Create a New Custom Metric

1. From the Oracle Management Cloud console navigation menu, select **Monitoring**, then **Monitoring Admin**, and finally **Custom Metrics**. The Custom Metrics page displays.
2. Click **Create**. The Create Custom Metric page displays.
3. Decide on a custom metric name. Be aware that both the **Metric Name** and **Display Name** must be unique across an entity type.
4. Enter the remaining general properties.
The selected adapter type defines the properties you must specify in Adapter Properties page. The following adapter types are available:
 - *OS Command Adapter*: Executes the specified OS command and parses each command output line (delimited by a user-specified string) into multiple values. The metric result is a mult-row, multi-column table.
 - *SQL Adapter*: Executes custom SQL queries or function calls against single instance databases and instances on Real Application Clusters (RAC).

 **Note:**

This applies to Oracle Databases only.

To create custom metrics based on SQL queries for MySQL and Microsoft SQL Server databases, see [Creating Custom Metrics for MySQL and SQL Server Databases](#).

- *JMX (Java Management Extensions) Adapter*: Retrieves JMX attributes from JMX-enabled servers and returns these attributes as a metric table.
5. Set the **Collection Schedule**. You defined the frequency with which metric data is collected by specifying collection schedule properties.
 6. On the Metric Columns page, you add metric columns defining the data returned from the adapter. From the **Add** menu, select **New metric column**. The Add Metric Column dialog displays.
Enter the required metric column properties and click **Submit** when you are done.

 **Note:**

The column order should match the order with which the adapter returns the data.

Compute Expressions (Advanced)

You use compute expressions to calculate the value of a metric column based on mathematical or logical operations performed on other metric columns. Compute expressions require at least one other metric column to be defined first, and can only include those metric columns that are listed before this metric column in order. You can use the up and down arrows to re-order metric columns.

The following table shows operators which can be used while defining compute expression.

Operator	Example	Explanation
+	Column1 + Column2	Returns the sum of the values of Column1 and Column2.
-	(Column1 + Column2) - Column3	First add Column1 and Column2 values, then subtract Column3 value and return the result.
*	(Column1*Column2) + Column3	First multiply Column1 and Column2 values, then add Column3 value and return the result.
/	(Column1 + Column2) / 2	Returns the average of Column1 and Column2 values.
__ceil	__ceil Column1	Returns the value of Column1 rounded off to the largest integer.
__floor	__floor Column1	Returns the value of Column1 rounded off to the lowest integer.
__round	__round Column1	This expression will round the value of Column1 to the nearest integer, away from zero.
==	Column1 == 1	Returns true if the value of Column1 is 1, else returns false.
!=	Column1 != 1	Returns false if the value of Column1 is 1, else returns true.
() ? : ;	(Status == 1) ? "UP": "DOWN"	This operator is equivalent to if then else statement. This expression will return "UP" if Status value is 1 otherwise it will return "DOWN"
__is_null	__is_null Column1	Returns true if the value of Column1 is NULL, else returns false.
__delta	__delta Column1	Returns the difference between the current value and the previous value of Column1.
__contains	Column1 __contains "ORA-"	Returns true if the value of Column1 contains the string "ORA-", else returns false.

Operator	Example	Explanation
__beginswith	Column1 __beginswith "ORA-"	Returns true if the value of Column1 starts with the string "ORA-", else returns false.
__matches	Column1 __matches "UP"	Returns true if the value of Column1 is equal to "UP", else returns false.
__length	__length Column1	Returns the length of string value of Column1.
__to_upper	__to_upper Column1	Returns the upper case of string value of Column1
__to_lower	__to_lower Column1	Returns the lower case of string value of Column1.
__interval	Column1 / __interval	Returns the Column1 value divided by the collection interval.

The value of the column is calculated using the given compute expression.

Usage:

This attribute specifies the formula for calculating the value of the column. Columns previously defined can participate in the calculation.

Refer to the examples for details about the expression grammar and usage.

Predefined special values:

- *__interval*: collect interval.
- *__sysdate*: current system time.
- *__GMTdate*: current GMT time.
- *__contains*: tests a given string expression for presence of a string expression.
- *__beginswith*: tests whether a given string expression begins with a specified string expression.
- *__endswith*: tests whether a given string expression ends with the specified string expression.
- *__matches*: tests whether a given string expression matches a specified string expression.
- *__delta*: computes the difference between the current value and the previous value.
- *__leadingchars*: returns the leading characters in the specified string.
- *__trailingchars*: returns the trailing characters in the specified string.
- *__substringpos*: returns the position of the occurrence of the pattern within a specified string.
- *__is_null*: tests whether the expression is NULL
- *__length*: returns the length of the string expression.
- *__to_upper*: converts the string to upper case.
- *__to_lower*: converts the string to lowercase.
- *__ceil*: returns the smallest integral value not less than identifier.

- `__floor`: returns the largest integral value not greater than the identifier.
- `__round`: rounds to nearest integer, away from zero.

Examples:

- `NAME="Average" COMPUTE_EXPR="(Column1 + Column2)/ 2"`

The value of the column is the average of the columns **Column1** and **Column2**.

- `NAME="Version" COMPUTE_EXPR="(Column1 __contains 'NetApp Release 7.') ? '7.X':'6.X'"`

The value of the column **Version** is computed as 7.X if column **Column1** contains the String NetApp Release 7..

- `NAME="Column1" COMPUTE_EXPR="(Column2 - Column3) "`

The value of the column **Column1** is the difference of the columns **Column2** and **Column3**.

- `NAME="Status" COMPUTE_EXPR="State __matches 'STARTED'"`

The value of the column Status is 1 if the value of column State matches the String STARTED and 0 otherwise.

- `NAME="Column1" COMPUTE_EXPR="(__is_null Column2)?'yes':'no'"`

The value of the column Column1 is yes if the value of column Column2 is null and no otherwise.

- `NAME="Source" COMPUTE_EXPR="((__length result) == 0) ? 'lanplus' : result"`

The value of the column Source is lanplus if the length of string value of column result is 0; el e it is the value of the column result.

- `NAME="Rate" COMPUTE_EXPR="(__ceil (Column1/__interval))"`

The value of the column Rate is the value of column Column1 divided by the collection interval, rounded up to the largest integer.

- `NAME="Column1" COMPUTE_EXPR="(Column2 == 0) ? 0 : ((Column3 / (Column2 / 8)) * 100.0))"`

The value of the column is the Column1 when Column2 and Column3 are existing metric columns.

- `NAME="PERCENTAGE_VALUE" COMPUTE_EXPR="(Column1 != 0) ? 100.0*(Column2/Column1) : 0"`

The value of the column is the total percentage of disk available where Column1 and Column2 are existing metric columns

Rate and Delta Metric Columns

You can create additional metric columns based on an existing data column that measure the rate at which data changes or the difference in value (delta) since the last metric collection. After at least one metric column has been created and the metric column row is selected in the table, two additional options appear in the **Add** menu:

- *Delta metric column on <selected metric column>*
- *Rate (per min) metric column on <selected metric column>*

To create a rate/delta metric column, click on an existing data column in the metric columns table and then select one of the rate/delta column menu options from the **Add** menu.

Usage Examples

- Add Delta metric columns based on another metric column

Example: You want to know the difference in the table space used since the last collection.

Delta Calculation:

```
current metric value - previous metric value
```

- Add Rate Per Minute metric column based on another metric column

Example: You want to know the average table space usage per minute based on the table space column metric which is collected every 1 hr.

Rate Per Minute Calculation:

```
(current metric value - previous metric value) / collection schedule
```

7. Once you're done, click **Review** to see your custom metric definition. If you need to change any properties or metric columns, go back to any of the previous pages and make the modifications.
8. Click **Submit** to create the custom metric. The new custom metric appears in the list on the Custom Metrics page.

Edit, Test, Delete or Publish of Your Custom Metric

After you click **Submit**, your custom metric is saved in *draft* status. When you select a draft status metric on the Custom Metrics page, you can edit, test, delete, or publish the custom metric. Also, the custom metric is only viewable by you, the custom metric owner.

When you're ready to push your custom metric to a production environment, it must first be published. You will no longer be able to edit or test the metric. You can only *deploy* the metric to production target entities at this point.

Edit a Custom Metric

1. Choose a custom metric in draft status from the list on the Custom Metrics page.
2. Click **Edit**. The custom metric property pages display.
3. Modify the custom metric properties.
4. Click **Submit** to save the changes.

Test Deploy a Custom Metric to One or More Entities

1. Choose a custom metric in draft status from the list on the Custom Metrics page.
2. Click **Test**. The Test Custom Metric page displays.
3. Click **Add Entity**. The Select Entities dialog displays.
4. Add one or more entities then click **Select**.
5. Click **Test** to begin the test deployment process. While the test deployment job is running, test status will be updated in the Test Results region.

IMPORTANT: DO NOT use your browser's refresh/reload function or navigate away from this page. Doing so will erase the test result data. You can set the auto-refresh rate by using the refresh drop-down menu.

Delete a Custom Metric

1. Choose a custom metric in draft status from the list on the Custom Metrics page.
2. Click **Delete**. The Confirmation dialog displays.
3. Click **Yes**.

Publish a Custom Metric

1. Choose a custom metric in draft status from the list on the Custom Metrics page.
2. Click **Publish**.

Deploy or Undeploy a Custom Metric

Once your custom metric has been published, you can deploy the metric to production target entities. If the custom metric has already been deployed, you have ability to undeploy the metrics from production target entities.

Deploy a Custom Metric to One or More Entities

1. Choose a custom metric in published status from the list on the Custom Metrics page.
2. Click **Deploy**. The Deploy Custom Metric page displays.
3. Click **Add Entity**. The Select Entities dialog displays.
4. Add one or more entities then click **Select**.
5. Click **Deploy** to begin the deployment process. While the deployment job is running, deployment status will be updated in the Deployment Results region.

IMPORTANT: DO NOT use your browser's refresh/reload function or navigate away from this page. Doing so will erase the deployment status data. You can set the auto-refresh rate by using the refresh drop-down menu.

Undeploy a Custom Metric from One or More Entities

1. Choose a custom metric in published status from the list on the Custom Metrics page.
2. Click **Undeploy**. The Undeploy Custom Metric page displays.
3. Select one or more deployed entities.
4. Click **Undeploy** to begin the undeployment process. While the undeployment job is running, its status will be updated in the Undeployment Status region.

IMPORTANT: DO NOT use your browser's refresh/reload function or navigate away from this page. Doing so will erase the undeployment status data. You can set the auto-refresh rate by using the refresh drop-down menu.

Delete a Published Custom Metric from the UI

After you have published a custom metric, if you later wanted to remove the custom metric, it was not possible to delete the custom metric from the UI. This release addresses this by allowing you to delete published custom metrics from the UI. You will be able to delete the published custom metric only if:

- You have been granted the OMC Administrator role.
- The custom metric has been undeployed from all entities.

Creating Custom Metrics for MySQL and SQL Server Databases

As discussed earlier, for Oracle databases, you define a Custom Metric using the SQL Queries for execution using a *SQL Query* collection method, which is not available for MySQL and Microsoft SQL Server databases.

To create a Custom Metric based on a SQL query for MySQL and Microsoft SQL Server databases, you can use the *OS Command* collection method with a specific OS Command that includes the SQL query. This is discussed in detail below.

1. From the Oracle Management Cloud console navigation menu, select **Monitoring**, then **Monitoring Admin**, and finally **Custom Metrics**. The Custom Metrics page displays.

The screenshot shows the Oracle Management Cloud console for Custom Metrics. It includes a header with the Oracle logo and 'MANAGEMENT CLOUD Monitoring'. Below the header, there's a section for 'Custom Metrics' with instructions and a 'Create' button. A table lists several custom metrics with columns for Name, Entity Type, Status, Deployed Entities, Display Name, and Description.

Name	Entity Type	Status	Deployed Entities	Display Name	Description
CustomMetric_DBVersion	Host (Windows)	Draft	0	CustomMetric_DBVersion	Test custom metric
Test_CM_DB	Oracle Database	Published	0	Test_CM_DB	test
pwd_check	Oracle Database	Published	0	pwd_check	
TEST_CM	Oracle Pluggable Database	Draft	0	TEST_CM	Custom metrics test
CustomMetric_Demo_MG	Microsoft SQL Server Database	Published	1	CustomMetric_Demo_MG	Custom Metric Group - Using OS Command to Run sql
CustomMetric_MG	Microsoft SQL Server Database	Published	1	CustomMetric_MG	Custom metric support for SQL Query execution using OS command
CustomMetric_Sample	Microsoft SQL Server Database	Draft	0	CustomMetric_Sample	Sample custom metric for SQL Server using OS command
DT2	Microsoft SQL Server Database	Published	1	DT2	One Row One Column

2. Click **Create**. The *Create Custom Metric: Basic Properties* page displays with the following properties:
 - Metric Group Name
 - Description
 - Metric Type (Performance or Configuration)
 - Entity Type (*MySQL* or *Microsoft SQL Server*)
 - Collection Method - **OS Command** (Only available option)
 - Collection Frequency

The screenshot shows the 'Create Custom Metric' wizard in Oracle Management Cloud. The 'Basic Properties' step is selected, and the following fields are visible:

- Metric Group Name:** Test_MetricGroup
- Description:** Metric Group for Testing
- Metric Type:** Performance
- Entity Type:** Microsoft SQL Server Database
- Collection Method:** OS Command
- Collection Frequency:** 15 Minutes

3. Select *Entity Type* as **MySQL** or **Microsoft SQL Server** and enter/select the **Basic Properties** parameters:
4. Click **Collection Method Properties** to go to the next page of the wizard.
5. In the *Collection Method Properties* page, in the **Command** field, you'll need to enter this specific command. (substitute `%STATEMENT%` with your SQL query for the Custom Metric).

```
%JAVA_HOME%/bin/java -classpath %PLUGIN_ROOT%/archives/*
oracle.sysman.emd.custommetric.CustomMetricQueryHelper "url=%url%"
"driver=%jdbcdriver%" "stmt=%STATEMENT%"
```

Note:

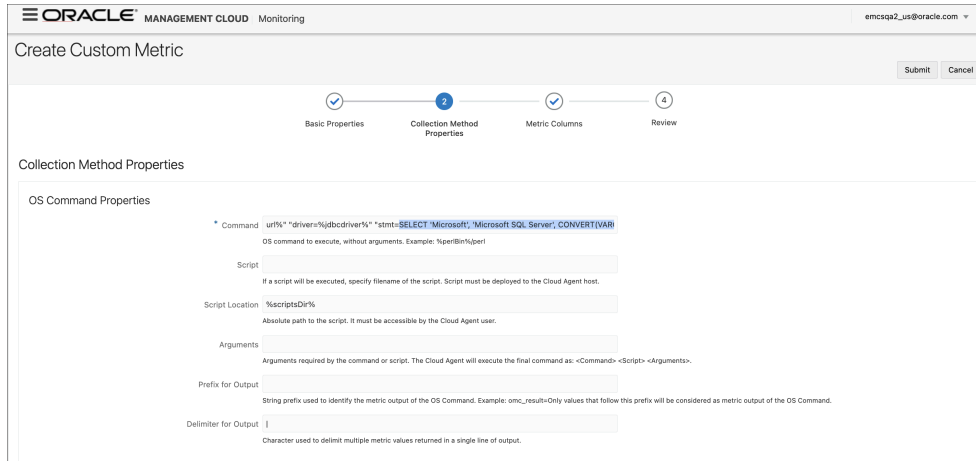
PL/SQL is not supported at this time.

Examples: SQL statements for SQL Server:

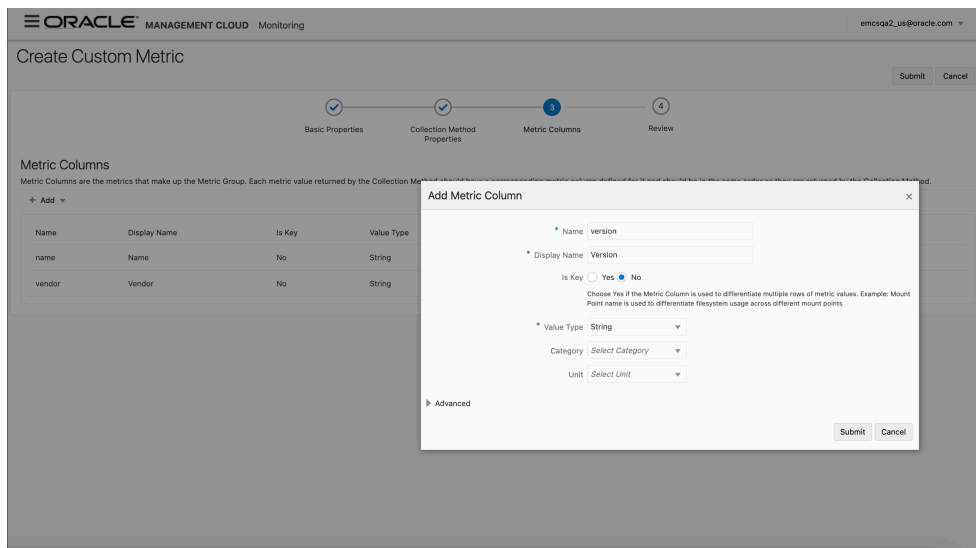
```
SELECT 'Microsoft', 'Microsoft SQL Server', CONVERT(VARCHAR(50),
SERVERPROPERTY('productversion'));
```

```
SELECT create_date FROM sys.databases WHERE database_id= DB_ID();
```

```
SET NOCOUNT ON DECLARE @DBInfo TABLE(DatabaseName VARCHAR(100),
SpaceAllocated FLOAT, SpaceUsed FLOAT) DECLARE @command VARCHAR(5000)
SELECT @command = 'Use [' + '?' + '] SELECT ' + ''' + '?' + ''' + ' AS
DatabaseName, CAST(sysfiles.size/128.0 AS FLOAT) AS SpaceAllocated,
CAST(CAST(FILEPROPERTY(sysfiles.name, ' + ''' + 'SpaceUsed' + ''' + ' )
AS FLOAT)/128.0 AS FLOAT) AS SpaceUsed FROM dbo.sysfiles' INSERT INTO
@DBInfo(DatabaseName,SpaceAllocated,SpaceUsed) EXEC sp_MSForEachDB
@command SELECT DatabaseName,SUM(SpaceAllocated)/1024 as
[SpaceAllocated],SUM(SpaceUsed)/1024 as [SpaceUsed] FROM @DBInfo GROUP BY
DatabaseName ORDER BY DatabaseName
```



6. Click **Metric Columns** at the top of page to go to the next page.
7. On the Metric Columns page, define the metric columns for each of the columns in the SQL query. To define each metric column, click **Add**. The *Add Metric Column* dialog displays.



Enter the required metric column parameters and click **Submit**. All columns added will appear in the *Metric Columns* table. For more information about metric column creation, see [Working with Custom Metrics](#) and [Custom Metric Collection Methods and Metric Columns](#).

8. Click on **Review** at the top of the page to go to the next page.
9. On the *Review* page, review the General Properties, Collection Schedule, Collection Method Properties and Metric Columns you defined in the previous step and, when ready, click **Submit**.

Basic Properties Collection Method Properties Metric Columns 4 Review

Review

General Properties

Metric Group Name	Test_MetricGroup
Display Name	Test_MetricGroup
Description	Metric Group for Testing
Metric Type	Performance
Entity Type	Microsoft SQL Server Database
Collection Method	OS Command

Collection Schedule

Collection Frequency	15 Minutes
----------------------	------------

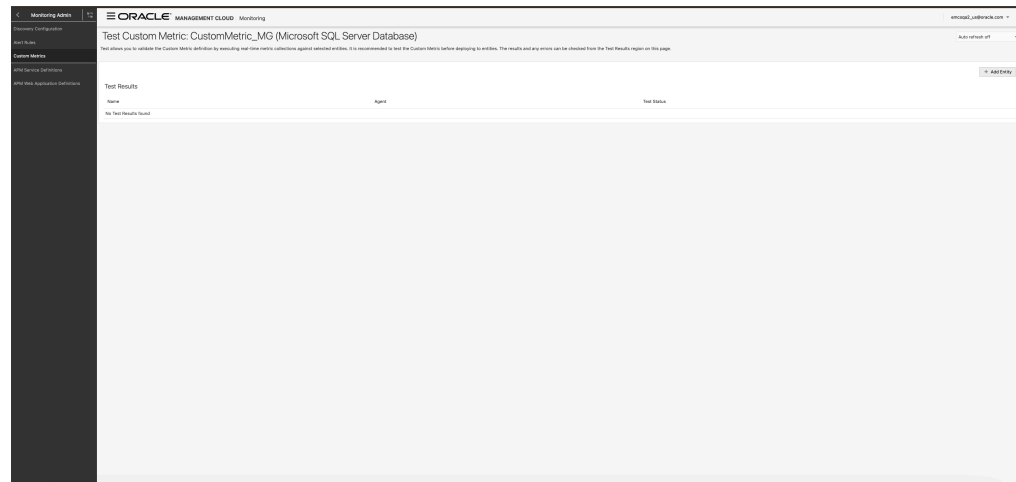
Collection Method Properties

Command	%JAVA_HOME%\bin\java -classpath %PLUGIN_ROOT%\archives\ oracle.sysman.emd.custommetric.CustomMetricQueryHelper "url=%url%" "driver=%jdbcdriver%" "stmt=SELECT 'Microsoft', 'Microsoft SQL Server', CONVERT(VARCHAR(50), SERVERPROPERTY('productversion'))"
Script Location	%scriptsdir%
Delimiter for Output	

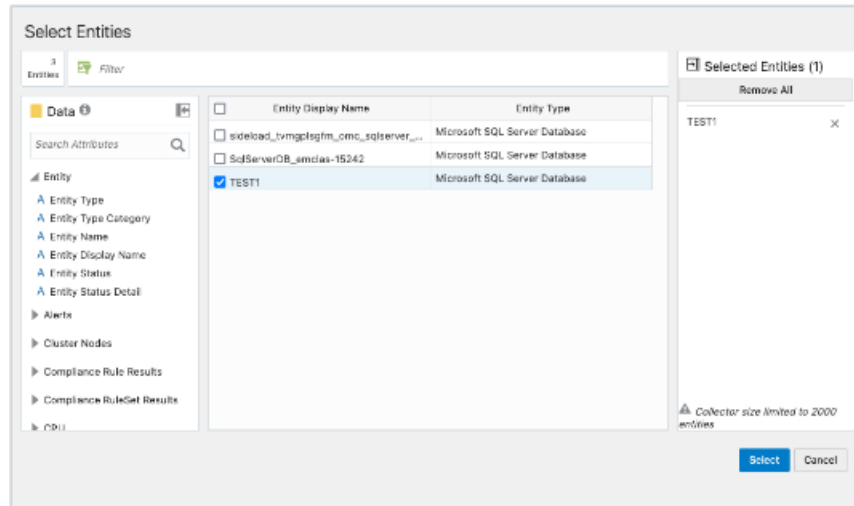
Metric Columns

Name	Display Name	Is Key	Value Type	Transient	Unit	Category
name	Name	No	String	No		
vendor	Vendor	No	String	No		
version	Version	No	String	No		

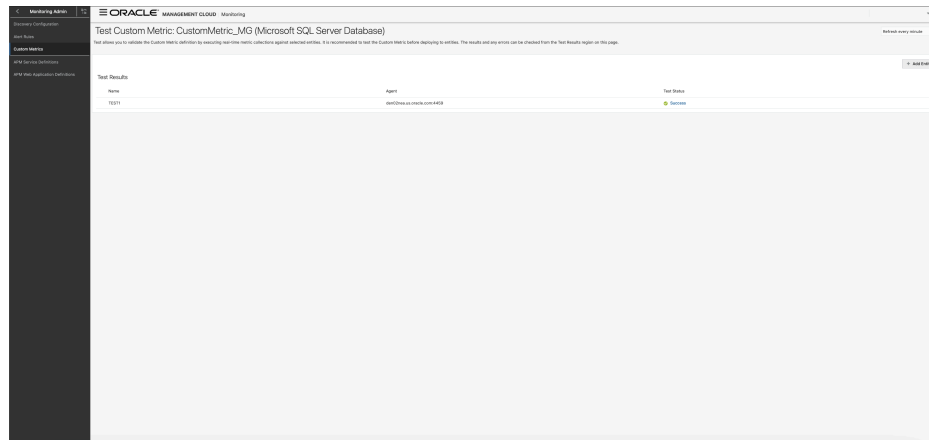
10. Test the newly defined Custom Metric against existing Microsoft SQL Server or MySQL Entity before Publishing and Deploying.
 - a. Choose a custom metric in draft status from the list on the *Custom Metrics* page.
 - b. Click **Test**. The Test Custom Metric page displays.



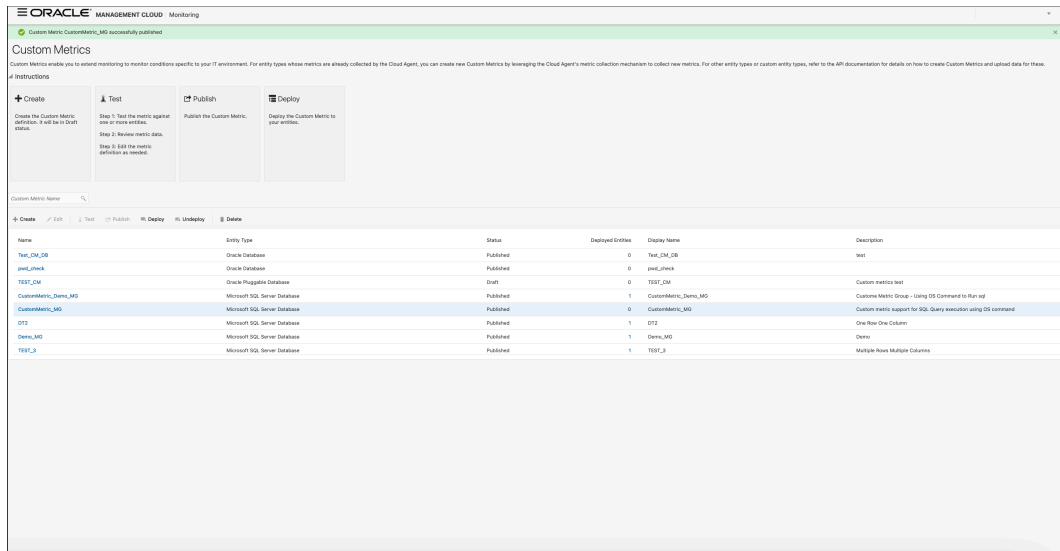
- c. Click **Add Entity**. The *Select Entities* dialog displays. Select a Microsoft SQL Server or MySQL entity.



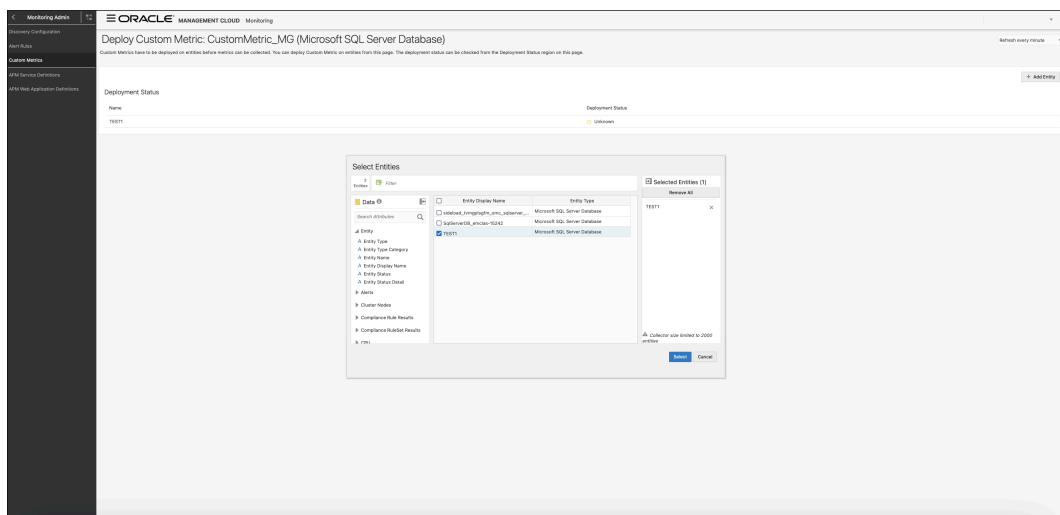
- d. Add one or more Microsoft SQL Server or MySQL entities then click **Select**.
- e. Click **Test** to begin the test deployment process. While the test deployment job is running, test status will be updated in the *Test Results* region.
IMPORTANT: DO NOT use your browser's refresh/reload function or navigate away from this page. Doing so will erase the test result data. You can set the auto-refresh rate by using the refresh drop-down menu.



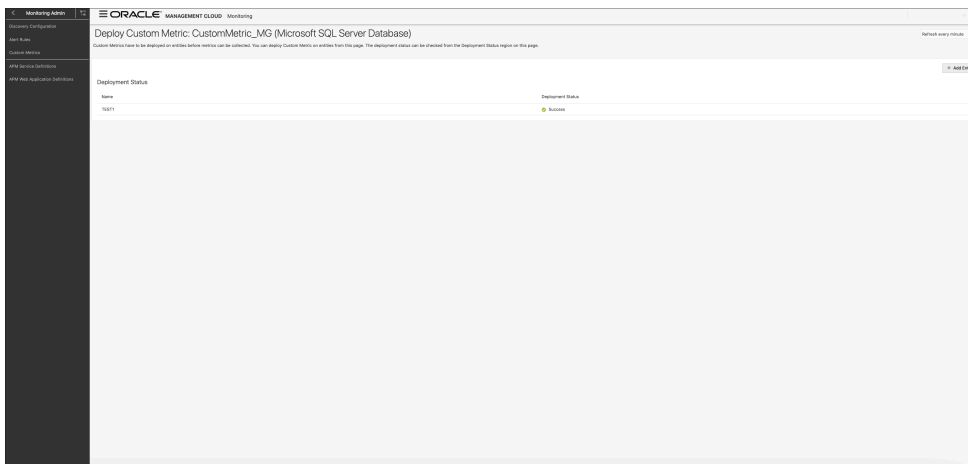
11. If the *Test* succeeds, publish the custom metric..



12. Deploy the Custom Metric against the Microsoft SQL Server or MySQL entities.



13. Verify that the deployment of the custom metric is successful.



14. After the deployment is completed, go to the homepage of the entity on which the custom metric has been deployed, where you will be able to add the Custom Metric to the charts in the homepage.

6

Host Process Monitoring

Process monitoring enables you to monitor the state and resource utilization of processes running on a host. This is useful in scenarios where you may want to keep a proactive eye on the state of the critical processes that make up an application. You can also monitor the CPU and Memory Utilization of these processes as well as get alerts if they cross boundary threshold conditions.

To identify the specific set of processes that need to be monitored, you define a Process Set, which is a user-specified set of processes. When defining a Process Set, you specify a name and a java Regex filter expression based on the attributes such as process name, process owner, or process command. For example, you can create a Process Set called "Java processes" which consists of all of the processes with 'java' in their name. All the processes that match the filter expression will be part of the Process Set.

The following metrics are collected for each Process Set:

- Process Id
- Process Name
- Process User
- Process State
- Memory Utilization
- CPU Utilization
- CPU Usage
- Virtual Memory Usage
- Physical Memory Usage
- Command and Arguments
- Count of processes in the Process Set

Process Monitoring Prerequisites

In order to monitor host process, ensure that you perform the following:

- Install a Cloud Agent on the host. See [Install Cloud Agents](#) for more information.

 **Note:**

For this release only UNIX and Windows platforms are supported.

- Install and configure a data transfer tool (such as cURL) to process REST API requests

Set Up Process Monitoring

Set Up and Use Process Monitoring

Step 1: [Create a Process Set](#)

Create a Process Set that identifies the set the processes that you would like to monitor.

Step 2: [Map the Process Set ID to One or More Hosts](#)

Identify the hosts on which the processes should be monitored.

Step 3: [Monitor the Processes from the UI](#)

You can now use the Oracle Infrastructure Monitoring UI to monitor your host processes as you would other metrics for the host.

Step 4: [Create Alert Rules to Monitor Process Status and Resource Consumption](#)

You can create alert rules on process metrics and send notifications when alerts trigger.

Create a Process Set

A Process Set represents a template for specifying host processes and consists of one or more filter patterns that specify a set of processes running on a host.

A host process gets matched to a Process Set if it meets all the filter criteria defined in the Process Set. A Process Set is created using REST APIs with a JSON payload. The JSON file contains the Process Set definition, which is constructed using regular expressions (regex) to filter out the processes that you're interested in. Once you've created the Process Set, you can then make REST API calls that perform process monitoring setup and configuration operations.

IMPORTANT: You must have access to the REST API for process monitoring. All process monitoring setup and configuration operations are performed using REST API calls. For this release, contact Oracle Support for the REST API documentation and reference MOS Note 2547614.1. Support will then provide you a zip file containing the REST API documentation.

Creating a Process Set

1. You first need to create a JSON file using the Process Set syntax specified in the process monitoring REST API. In this file, you'll specify the following:
 - Name of the Process Set to be displayed (as a metric) in the UI.
 - A brief description of the Process Set.
 - A *regular expression* (Regex) search pattern that will be used to filter out only those processes of interest. For example, if you want to filter out host Java processes, you could define the following search pattern:

```
^java$
```

For comprehensive information on creating Regex expressions, see "[Summary of regular-expression constructs](#)" in the Java API.

2. Create the Process Set using the REST API and JSON file you created in the first step. You can use any URL-based data transfer tool, such as cURL, to make the REST API call. See [Infrastructure Monitoring Cloud Service Master Note \(Doc ID 2195015.1\)](#) in My Oracle Support (MOS) for access to the REST API for process monitoring.
Upon successful execution, the API call returns the pertinent Process Set definition information.

- Process Set Name
- Process Set meID
- Execution confirmation message.

In the following example, the returned output shows that the Process Set named `Java Processes` with the meID `8A978264528271548A80BB656B526FCA` was added successfully.

```
{"name": "Java Processes", "meID": "8A978264528271548A80BB656B526FCA", "message": "Successfully added Process Set : Java Processes"}
```

Important: You'll need the Process Set meID (shown in the REST API output above) for the next step when you map this Process Set to the host whose processes you want to monitor.

Map the Process Set ID to One or More Hosts

Once you've created the Process Set and obtained the Process Set meID, you must now associate it with the hosts on which the processes of interest are running.

Note:

You must have the meID of the host in order to complete this step. The host meID can be obtained using the extended entity REST APIs. For this release, the REST API documentation can be accessed via the [Infrastructure Monitoring Cloud Service Master Note \(Doc ID 2195015.1\)](#) in My Oracle Support (MOS).

Map the Process Set to the Host

1. Create a JSON file using the map file syntax specified in the process monitoring REST API. In this file, you'll specify the following:
 - meID of the Process Set
 - Name of the Process Set
 - meID of the host whose processes you want to monitorThe meID can be obtained using the Oracle Management Cloud extended APIs.

Note:

See *Oracle Management Cloud: Find the meID of a Host Entity Using REST API* in the References section of the [Infrastructure Monitoring Cloud Service Master Note \(Doc ID 2195015.1\)](#).

2. Map the Process Set to the host using the process monitoring REST API. Upon successful execution, the API call returns the Process Set-Host mapping status.

In the following example, the returned output shows that the execution of the mapping API call was successful (`mappingResponse` is 200) and the execution ID (`ecid`).

```
{ "mapping": { "mappingResponse": 200, "mappingMessage": "Mappings created successfully.", "ecid": "a1753f33-93a3-bf2f-fc946c26" } }
```

Monitor the Processes from the UI

Once you've mapped your Process Set to a host, you can monitor metrics for the host processes from the host entity homepage.

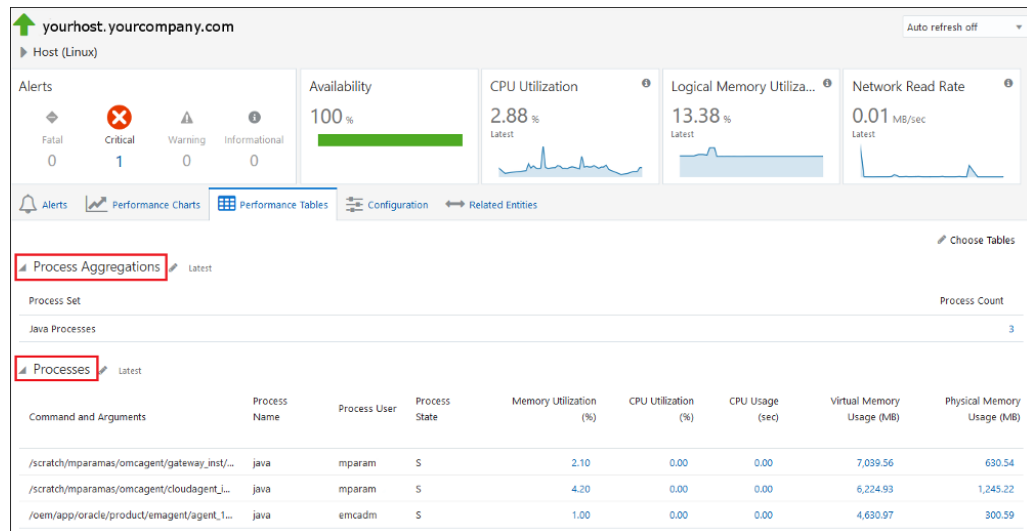
On the host homepage, Process Set metrics appear in the **Performance Tables** and **Performance Charts** tabs.

Performance Tables Tab

In the Performance Tables tab, performance set metrics appear under two metric groups:

- *Process Aggregations*: Running count of processes defined in the Process Set.
- *Processes*: State, CPU, Memory usage of each process matching the filter expression defined in the Process Set.

Figure 6-1 Performance Tables



Process States

The current state of a process is shown in the Process State column. Possible state values are shown in the following table.

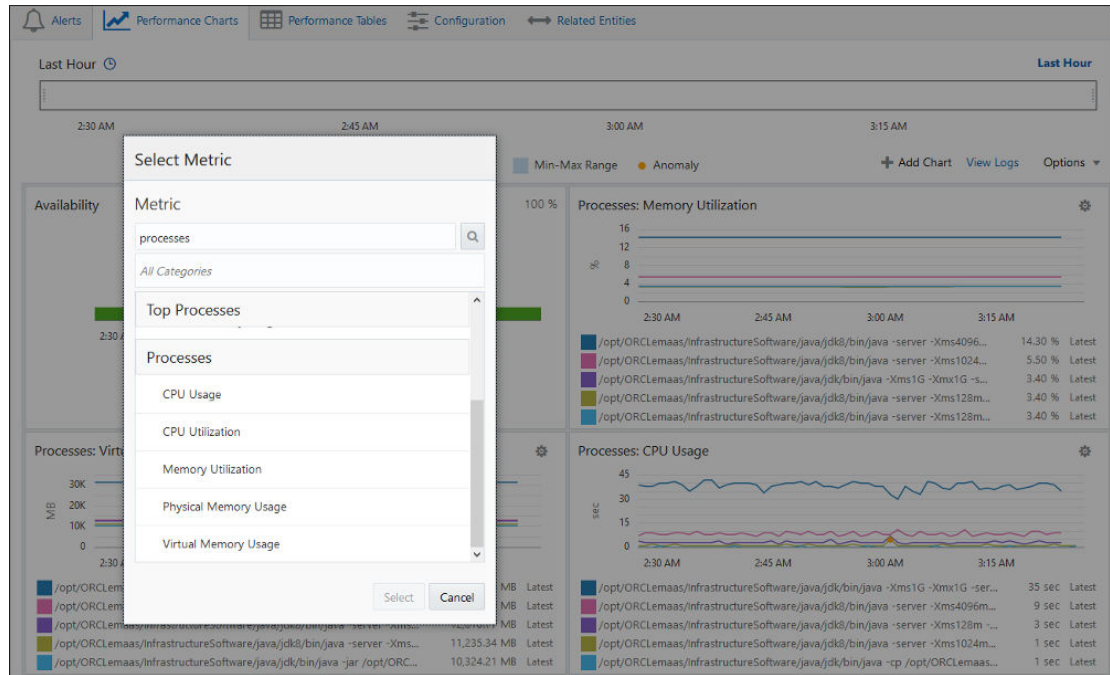
State	Description
D	Uninterruptible sleep (usually IO)
R	Running or runnable (on run queue)

State	Description
S	Interruptible sleep (waiting for an event to complete)
T	Stopped, either by a job control signal or because it is being traced.
W	Paging (not valid since the 2.6.xx kernel)
X	Dead
Z	Defunct ("zombie") process, terminated but not reaped by its parent.

Performance Charts Tab

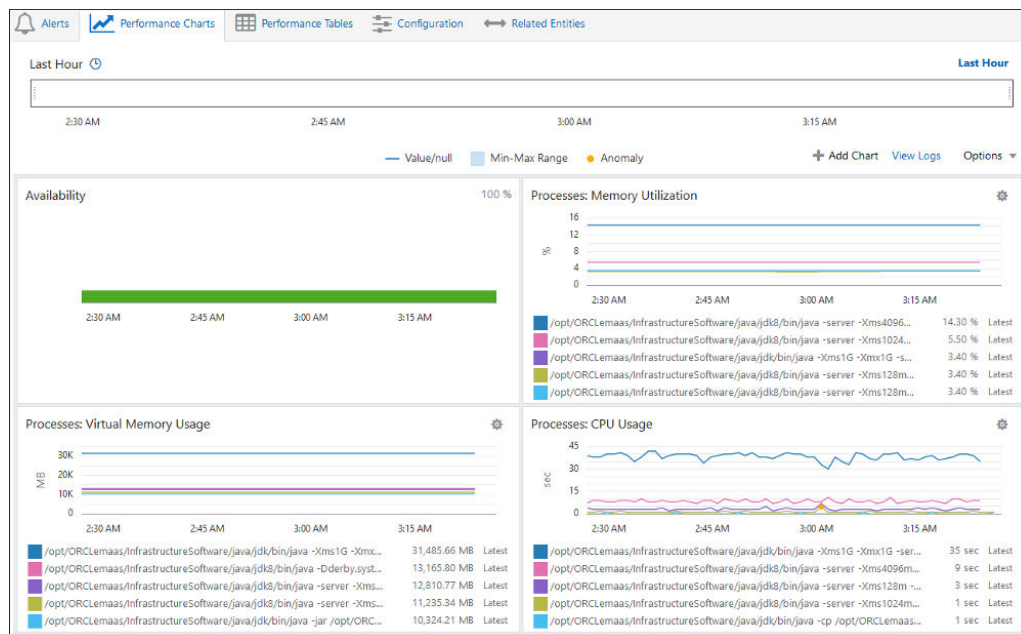
You can add any of the process metrics such as CPU Usage, CPU Utilization, Memory Utilization, etc. to the Performance tab. When adding a chart, you can type "process" into the search field to display the process metrics. Once added, the chart will show the processes with the highest value of the metric chosen. For example, if you chose CPU Usage, it will show you the top five processes in the Process Set with the highest CPU Usage.

Figure 6-2 Available Performance Chart Process Metrics



Because of the potentially large number of processes, only the top five process are shown in each chart.

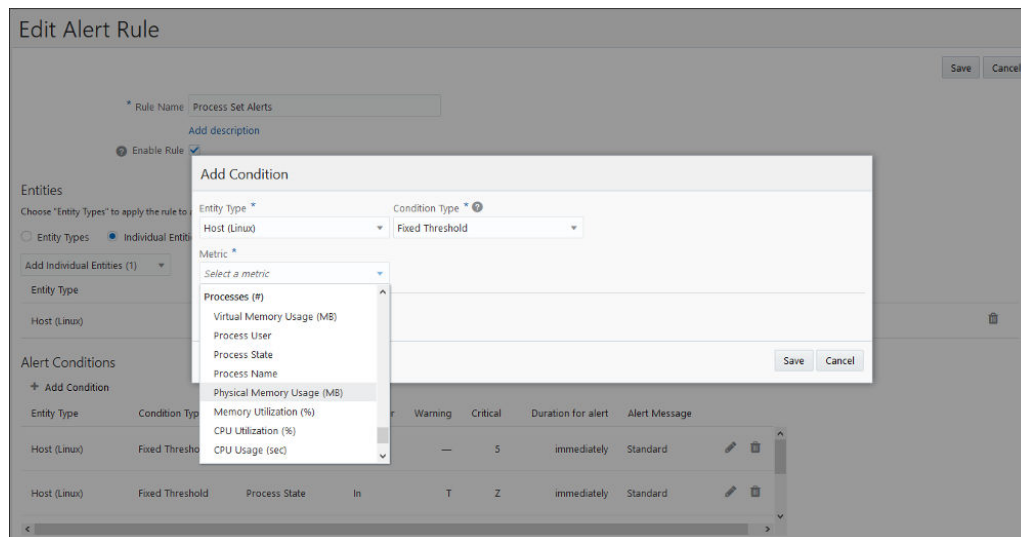
Figure 6-3 Process Performance Charts



Create Alert Rules to Monitor Process Status and Resource Consumption

You can create alert conditions based on any of the process metrics for both Process Aggregation and Processes metric groups.

Figure 6-4 Add an Alert Rule Condition



Alert condition properties will change depending on the process metric selected. The following graphic illustrates an alert condition where if any process in the Java

Processes process has CPU utilization greater than 80 for at least 30 minutes, a warning alert is generated. If CPU utilization is greater than 90 for at least 30 min, then a critical alert is generated.

Figure 6-5 Alert Condition Properties

The screenshot shows the 'Add Condition' dialog box. At the top, 'Entity Type' is set to 'Host (Linux)' and 'Condition Type' is 'Fixed Threshold'. The 'Metric' is 'CPU Utilization (%)'. Below this is a table for defining conditions:

Condition	Operator	Warning	Critical
Process Set: Java Process	>	80	90

Below the table, it says 'Generate an alert when the metric is outside the specified threshold for 30 minutes'. There are 'Save' and 'Cancel' buttons at the bottom right.

The following graphic illustrates an alert condition where a critical alert is generated immediately if a process does not exist (process count is less than one).

Figure 6-6 Alert Properties: Process No Longer Exists

The screenshot shows the 'Add Condition' dialog box. 'Entity Type' is 'Host (Linux)' and 'Condition Type' is 'Fixed Threshold'. The 'Metric' is 'Process Count'. Below this is a table for defining conditions:

Process Set	Operator	Warning	Critical
Equals: Java Processes	<	value	1

Below the table, it says 'Generate an alert when the metric is outside the specified threshold immediat...'. There are 'Save' and 'Cancel' buttons at the bottom right.

For more information on creating alerts, see [Create an Alert Rule](#).

For more information on setting up notification channels, see [Set Up Notification Channels](#).

7

Set Up Alert Rules

Alert rules allow you to set up metric thresholds and define how to be notified if an entity is down or metrics thresholds are exceeded.

You must have Oracle Management Cloud Administrator privileges to set up Alert Rules.

Topics

- [Typical Workflow for Setting Up Alert Rules](#)
- [Set Up Alert Thresholds and Notifications](#)

Typical Workflow for Setting Up Alert Rules

Set up Alert Rules for your monitored infrastructure to trigger alerts on impending issues.

Table 7-1 Typical Workflow for Setting Up Alert Rules

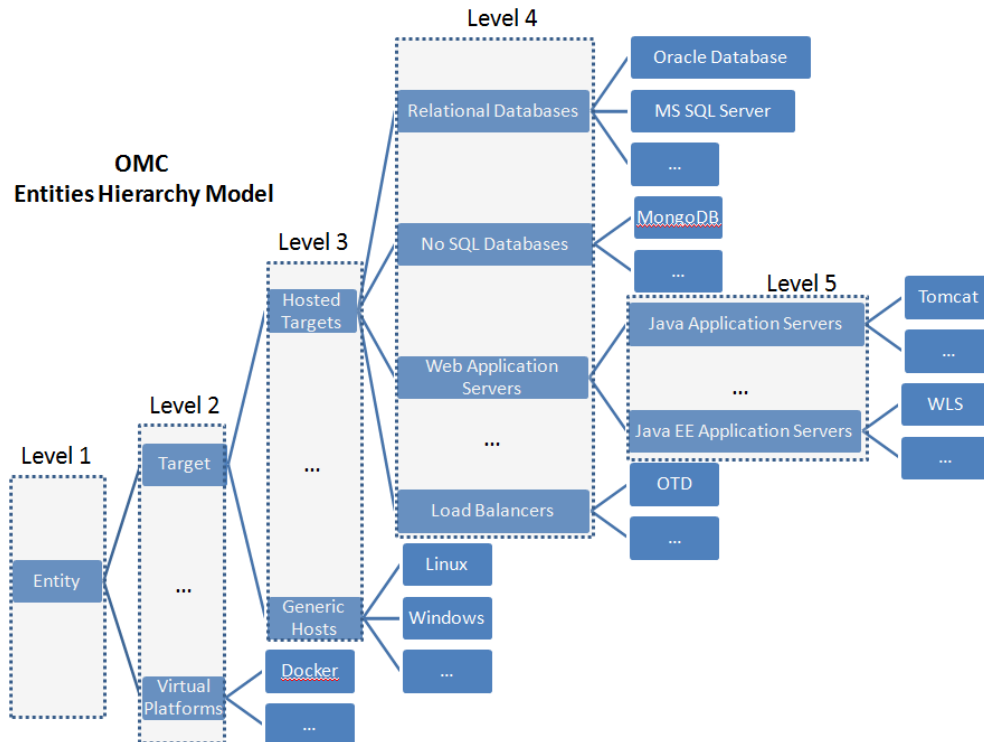
Task	Description	More Information
Select the entities for which you want to set up alerts.	Choose individual entities or entity types to set up alerts on.	Set Up Alert Thresholds and Notifications
Specify the alert conditions and notification options.	Select threshold conditions that apply to your entities in order to monitor performance. Select your notification options for performance and availability alerts.	Set Up Alert Thresholds and Notifications

For a brief video demonstrating how to create an Alert Rule, see [Set Up an Alert Rule](#).

Set Up Alert Thresholds and Notifications

Oracle Management Cloud (OMC) collects performance and availability metrics for all entities set up for monitoring. The *alerts* sub-system informs you of availability or performance problems. *Alert Rules* enable you to define how alerts are triggered (if they are not automatic) and how you get notified.

Setting up alerts thresholds or notifications requires an understanding of how entities are defined in Oracle Management Cloud. Each entity is defined using a multi-level hierarchical model as pictured below.



Each entity or entity type is defined by a set of characteristics, it has a parent and may have other children. For example, a *Generic Host* is an operating system (OS) independent *Target* and it has children entities that are specific OS Hosts, like *Linux*, *Windows* and so on. . The metrics collection functionality takes advantage of this model so each monitored entity has entity-specific metrics as well as metrics inherited from each level it descended from. For example, OMC collects metrics at level 3 that are common to all *Generic Hosts*, independent of the vendor. A *Linux Host*, since its parent is a *Generic Host*, inherits all the metrics collected for *Generic Hosts* and its ancestors, as well as *Linux* specific ones, if any.

Availability problems (for example, an entity is down, or agents or hosts are unavailable) are *automatically* detected by Oracle Infrastructure Monitoring and availability alerts are *automatically* generated. *Performance* metrics deviations, however, can be monitored by setting up Alert Rules. In addition, Alert Rules allow you to specify how to be notified when alerts are triggered.

Create an Alert Rule

To set up Alert Rules for your enterprise, you must be logged in to the services as an *Oracle Management Cloud Administrator* user. Navigate to the main service menu, go to the **Alert Rules** page and select **Create Alert Rule**.

Alert Rules

To set up Alert Rules for your enterprise, you must be logged in to the services as an *Oracle Management Cloud Administrator* user. Navigate to the main service menu, go to the **Alert Rules** page and select **Create Alert Rule**.

1. Choose a sample rule if any of the sample rules provided seem appropriate for your environment. Sample rules help you populate the rule definitions with values typical for a particular entity type.

2. Enter a **name** and optionally a **description** for your new alert rule.
3. Alert rules can apply to one or more entity *types* (based on the entities definition model described above), groups, or they can apply to an *individual* entity.

You can choose one of these options:

- **Entity Types** if the rule will apply to a set of entities of the same type. This includes specific entity types (for example, MySQL database or Tomcat, which are entities with no children in the definition model above) or generic entity types. Here are some examples of generic entity types:
 - A Relational Database, an entity type with several children in the definition model above; choose this to indicate that this rule applies to all entities that have the Relational Database as a parent.
 - A Hosted Target; choose this to indicate that this rule applies to all entities that run on a host, such as databases and application servers.
 - A Target; choose this if the alert rule will apply to all Hosted Targets and Generic Hosts entities and their children.
 - Any other custom defined entity type.
- **Individual Entities**, if the rule will apply to specific entities defined in your monitoring environment.

Your selection will determine the options that follow in selecting alert conditions.

Best Practice Tip: As mentioned above, alert rules can also apply to groups or composite entities. You can have an alert rule automatically apply to specific entities, by specifying a group, such as a dynamic group, as the entity for the alert rule. As more entities are monitored, you can add them to the group and the alert rule will automatically apply. You can further streamline this by creating dynamic groups based on tags. When you add an entity and also add the appropriate tag, the entity automatically joins the dynamic group and thus be part of the alert rule.

4. Specify the alert conditions. *Availability* monitoring is based on the availability of an entity. For *performance* monitoring, alert conditions are based on set thresholds of various performance metrics. Select one of the following then click **Save** to save these settings:
 - **Fixed Threshold** condition to generate an alert when a metric exceeds a set metric threshold value. From the metric drop-down, choose the metric of interest and the operator. Use the graphical display (chart) of historical values of that particular metric to guide you in entering the *Warning* and *Critical* values. Hover over the graph to see the metric values at various points in time. If metrics are associated with key values, you can set individual warning and critical thresholds values for each key value. For example, the File System Space Available % host metric has mount points as key values. You can set warning and critical thresholds for each mount point defined on that host.

The alert will be triggered immediately, unless you specify the number of minutes that the value should remain outside the threshold before an alert is generated.

You can also use a statistical value of the metric (Average, Sum, Minimum, Maximum, Count) as the value to be used when evaluating against specified metric thresholds. For example, if CPU utilization % has a warning threshold of 70% and a critical threshold of 90%, you can specify that when the Average CPU utilization value remains over 70% for more than 30 minutes, generate an alert.

You can select a standard alert message or, select to customize the message when the alert first triggers as well as the message sent when the alert clears. Use the

provided tokens, encapsulated as `{token_name}`, to construct a meaningful custom alert message. This table lists the tokens you can use:

Table 7-2 Valid Tokens

Token Name	Description
sys.entityDisplayName	Name of the entity
sys.entityId	Entity ID
sys.value	Value of the metric that triggered the alert
sys.criticalThreshold	Critical threshold used in evaluating the alert condition
sys.warningThreshold	Warning threshold used in evaluating the alert condition
sys.operator	Operator used in evaluating the alert condition
sys.conditionName	Alert condition name
sys.ruleName	Alert Rule name
sys.slidingWindowSize	Number of minutes that the metric crossed the threshold before the alert was triggered
sys.defaultMessage	<p>Show the default message normally generated for the alert condition. Additional text can be added before and after this token. The additional text can include other alert message tokens.</p> <p>Example</p> <pre>"messageTemplates": { "warning": "Warning Default message : % {sys.defaultMessage} %. Please contact the administrator to report the issue for entity % {sys.entityDisplayName}%. "}</pre> <p>Generated Message</p> <p><i>Warning Default message : TestMetric1_num for TestMsgTempEntity is 11; it is greater than expected value of 10 . Please contact the administrator to report the issue for entity TestMsgTempEntity.</i></p>

For a complete list and descriptions of all metrics collected for each entity type, see List of Supported Entities in *Metric Reference for Oracle Infrastructure Monitoring*.

Alert Condition Notes

When creating an alert condition as part of an Alert Rule, you can specify *notes* that contain runbook instructions (or a runbook URL) that should be followed when responding to an alert.

- **Other Types of Alert Conditions**

In addition to alerting for specific metric thresholds, you can also have Oracle Infrastructure Monitoring send alert notifications based on the following conditions:

Alert Anomalies: Alerts can be triggered when metric threshold values occur outside the bounds of expected *normal* values. Oracle Infrastructure Monitoring determines what is *normal* behavior for a monitored entity by creating metric baselines. A baseline is an expected range of values for a

metric based on its past performance. Baselines are automatically tracked and adjusted for seasonality, or the ability to adjust baseline values according to the time of day, or day of week once there is sufficient historical metric data.

When monitoring for anomalies, you may not want to raise an alert every time an anomalous condition occurs. Instead, you may only want to raise an alert if a metric anomaly occurs for a specific duration. For example, you don't care if there is a brief spike in a host's CPU usage, but you do want to be alerted if CPU usage remains abnormally high for 5 minutes or more. When defining an Alert Condition, you can specify duration-based anomalies.

Early Warning: Early warning conditions will trigger an alert when, based on historical metric data, Oracle Infrastructure Monitoring predicts the provided alert threshold will be crossed in the future.

- **Availability** alerts are automatically raised by Oracle Management Cloud when entities are detected to be *Down* or *Not Heard From* in the case of hosts, agents, gateways and data collectors. *Down* alerts or *Not Heard From* alerts on host and agent entities are alerts of **Fatal** severity. To get notifications for alerts of **Fatal** severity, choose the **Availability** alert condition.
5. Specify the recipients for the alert notifications by choosing the appropriate notification channel. See [Set Up Notification Channels](#) for more information.
- Customizing When Notifications are Sent:* By default, notifications are sent when alerts are first raised, when they worsen in severity, or when they are closed. If the default notification settings don't fit your needs, you can change them by setting various options under *Advanced Options*. You can change:

- The alert conditions under which a notification should be sent (use default or on every severity change).

 **Note:**

This option only applies to Email, Mobile, and Slack notification channels.

- The alert conditions under which a notification should be suppressed (based on selected severities).

 **Note:**

This option only applies to Email, Mobile, and Slack notification channels.

- Whether or not a notification should be sent repeatedly. Repeat notifications allow administrators to be notified at specified intervals until an alert is either cleared or acknowledged or the number of *Maximum number of repeat notifications* has been reached.

 **Note:**

This option only applies to Email notification channels.

6. Click **Save** to save the alert rule.

Once the alert rule has been created, it appears in the table on the Alert Rules page where it can be enabled, disabled, or deleted. Disabled alert rules do not create alerts or send notifications. Alert rules can also be enabled/disabled while editing/creating the alert rule.

Set Up Notification Channels

When an Oracle Management Cloud alert is raised, worsens in severity, or clears, you may want to be actively notified through email, by push notifications (mobile devices), or have a third-party application take action. These types of notifications are defined using *notification channels*.

Types of Notification Channels

Classes of notification destinations are called *notification channels*. Notification channels allow you to set up and reuse functional groups of notification recipients, such as regional administrators, IT managers, or other Web servers without having to specify large numbers of individual destinations repeatedly. Once you set up a notification channel, you can reuse the channels across different alert rules.

Available of notification channels:

- **Email:** Allows you to manage/logically organize the email recipients of alert notifications. For example, you can create an email channel for the *On-Call Team* containing the email addresses of the IT operators that staff the on-call team, or a *DBA Team* email channel containing all email addresses of database administrators in your department.
[Create an Email Notification Channel](#)
- **Mobile:** Allows you to send push notifications to one or more mobile phones associated with Oracle Management Cloud users.
[Create a Mobile Notification Channel](#)
- **Integration (WebHook):** Allows you to send HTTP POST (WebHook) messages to a destination URL. A WebHook notification message contains a JSON payload that specifies pertinent information to a destination Web application.
[Create a WebHook Notification Channel \(Integration\)](#)
- **PagerDuty:** Allows you to integrate Oracle Management Cloud's monitoring capability with PagerDuty's incident response services.
[Create a PagerDuty Notification Channel](#)
- **ServiceNow:** Allows you to integrate Oracle Management Cloud's monitoring capability with ServiceNow's incident management services.
[Create a ServiceNow Notification Channel](#)
- **Slack:** Allows you to send Oracle Management Cloud notifications to a specified Slack channel.
[Create a Slack Notification Channel](#)

Create an Email Notification Channel

Notification channels can be defined while creating/editing an alert rule from the *Alert Rules* UI or via the *Notification Channel* UI.

To create an email notification channel from the Notification Channel UI:

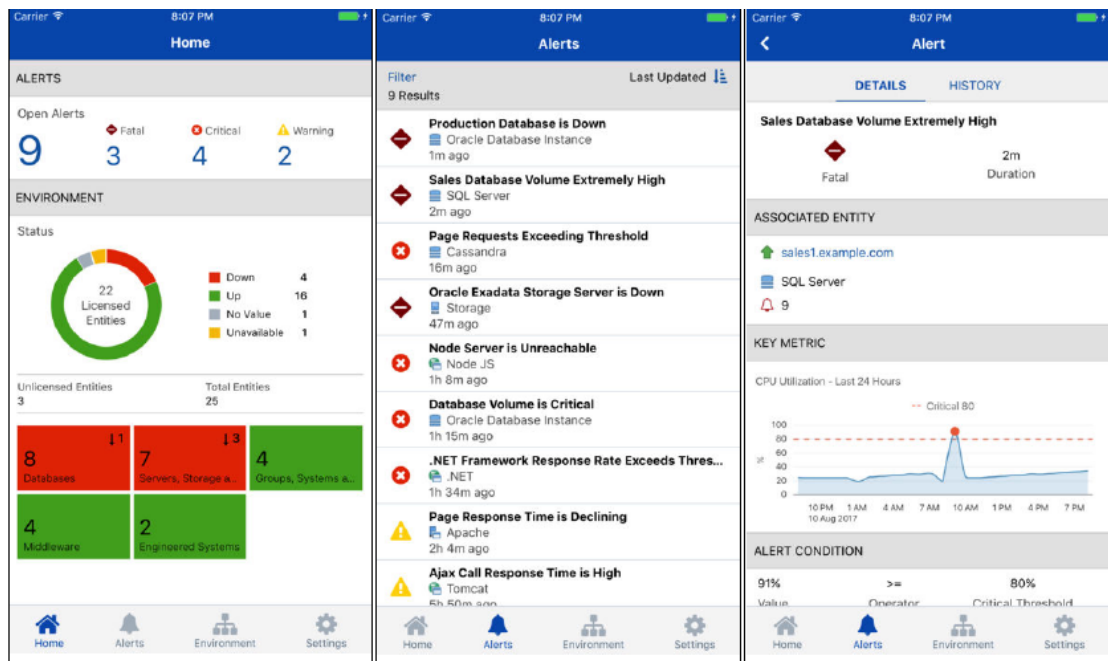
1. From the Management Cloud side menu, select **Administration**, and then **Notification Channels**.
2. On the Notification Channels page, click, **Create Notification Channel**. The four notification channel options are displayed.
3. Click on the desired notification channel type and specify the requisite information.

The most common method of notification is via email. Using an email notification channel lets you specify the channel and rather than having to specify individual emails every time. For an email notification channel, you specify:

- **Channel Name:** An intuitive name for this notification channel.
- **Email Addresses:** A comma-separated list of recipient email addresses .
- **Email Language:** The language that will be used for the date on which the alert was raised
- **Email Timezone:** The timezone that will be used for the time on which the alert was raised

Create a Mobile Notification Channel

In order to receive push notification on your mobile device, the Oracle Management Cloud Mobile application must be installed on your device and signed into. As shown below, the Oracle Management Cloud Mobile application offers anywhere access to information about the entire IT infrastructure managed by Oracle Management Cloud.



1. From the Management Cloud side menu, select **Administration**, and then **Notification Channels**.
2. On the Notification Channels page, click, **Create Notification Channel**. The four notification channel options are displayed.

3. Click on the desired notification channel type and specify the requisite information. For a Mobile notification channel, you specify:
 - **Channel Name:** An intuitive name for this notification channel.
 - **OMC User Names:** A comma-separated list of recipient Oracle Management Cloud users.

Create a WebHook Notification Channel (Integration)

In addition to notifying people, Oracle Management Cloud can also send relevant information to third-party Web applications (such as Slack or Hipchat) when an alert is raised, thus allowing you extend Oracle Management Cloud functionality by having third-party applications carry out actions in response to an Oracle Management Cloud alert notification. This type of system integration is achieved using WebHooks (HTTP POST message containing a JSON payload that is sent to a destination URL).

1. From the Management Cloud side menu, select **Administration**, and then **Notification Channels**.
2. On the Notification Channels page, click, **Create Notification Channel**.
3. Click on the desired notification channel type and specify the requisite information. To define a WebHook notification channel, you need to specify:
 - **URL:** Destination URL where the JSON payload is sent. If you want to specify a port number, please note that only port numbers 80 and 443 are supported.
 - **Authentication Type:** Type of authentication used when accessing the destination URL. You can specify **None** or **Basic** (Username and Password). When specifying Basic credentials, you choose whether you want to use *Existing Credentials*, or create *New Credentials*. For every new credential, you specify a *Credential Name* that will be used for future identification. If you select the *Existing Credentials* option instead of *New Credentials*, the *Credential Name* will appear as a selectable item in the drop-down list.
 - **Setting HTTP Header and Payload Values**

A field value can be a fixed string or token (i.e., payload variable) or combination of both.

When the value is a combination of fixed strings and tokens, the following rules must be followed.

- Value should start with `${`
- Value should end with `}`
- Each fixed string should be enclosed inside a pair of single quotes.
- Token names should be entered without enclosing them inside `${` and `}`
- To enter consecutive multiple tokens, place a pair of single quotes between each pair of tokens for separation, e.g.,
`alert.id'alert.message.`

Examples:

```
`${Alert ID: 'alert.id'}  
`${Alert ID: 'alert.id' Alert Message: 'alert.message'}  
`${Two tokens with a space between them: 'alert.id'  
'alert.message}
```

```

${'Two tokens without any space between them:
'alert.id'alert.message}

```

- **HTTP Headers:** Headers to be included with the message. For each header, value can be a fixed string, payload variable, combination of both. When the value is a combination of fixed strings and payload variables, follow the conventions listed in the previous bullet (*Setting HTTP Header and Payload Values*).
- **Payload:** The JSON payload specifying the requisite information sent to the destination URL. Payload consists of the following JSON code:

```

{
  "alertId": "${alert.id}",
  "ruleName": "${rule.ruleName}",
  "conditionName": "${rule.conditionName}",
  "updateType": "${updateType}",
  "message": "${alert.message}",
  "severity": "${alert.severity}",
  "time": "${alert.time}",
  "eventName": "${alert.eventName}",
  "alertDetailUrl": "${alert.detailUiUrl}",
  "entityId": "${entity.id}",
  "entityName": "${entity.name}",
  "entityType": "${entity.type}",
  "entityDisplayName": "${entity.displayName}",
  "entityHostName": "${entity.hostName}"
}

```

The JSON payload provides 14 tokens that are used to pass information to the destination URL. You can modify the JSON payload to send specific information required by the receiving Web application.

Token Name	Description
<code>\${alert.id}</code>	ID of the alert. Example: 1234
<code>\${updateType}</code>	Type of update. Example: created, updated, closed
<code>\${rule.ruleName}</code>	Name of the alert rule that triggered the alert.
<code>\${rule.conditionName}</code>	Name of the condition triggering the alert.
<code>\${alert.eventName}</code>	Name of the event.
<code>\${alert.message}</code>	Text of the alert message.
<code>\${alert.severity}</code>	Severity of the alert. Example: warning, critical
<code>\${alert.time}</code>	Date/time when the update occurred Example: 2017-07-05T20:22:04.134Z
<code>\${alert.detailUiUrl}</code>	URL where explicit details about the alert can be found.
<code>\${entity.id}</code>	ID of the entity.
<code>\${entity.name}</code>	Name of the entity impacted (internal name).
<code>\${entity.displayName}</code>	Display name of entity impacted.
<code>\${entity.type}</code>	Type of the entity. Example: APM Server Request, host(linux)
<code>\${entity.hostName}</code>	Host name of the host where the entity resides.

Token Name	Description
<code>\${entity.shortHostName}</code>	Shortened host name. This name does not include the domain name.

Create a PagerDuty Notification Channel

When an Oracle Management Cloud alert is raised, you can have that alert sent to PagerDuty for incident management. Oracle Management Cloud updates the same ticket when the alert is updated by PagerDuty, and closes the ticket when PagerDuty clears the alert. Integrating Oracle Management Cloud with PagerDuty is a two-phase process: Configure PagerDuty to receive Oracle Management Cloud alerts and then create a PagerDuty notification channel.

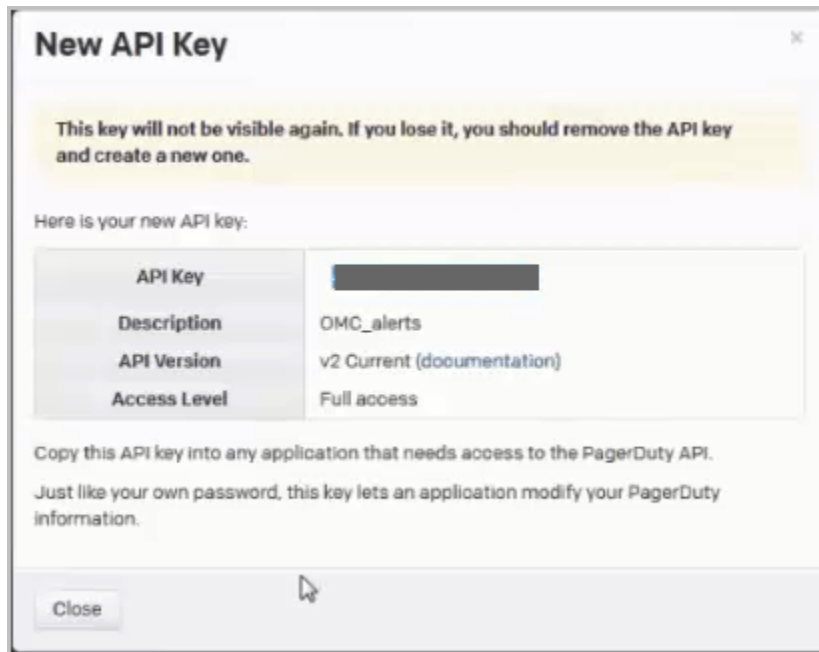
Obtaining the PagerDuty Authorization and Integration Keys

You need to obtain both an Authorization Key and an Integration Key from PagerDuty before you can define the PagerDuty notification channel.

1. Go to the PagerDuty authentication page and log in.
2. From the PagerDuty Configuration menu, select **API Access**. The API Access Keys page displays.
3. Click **Create New API Key**. The Create API Key dialog displays.



4. Enter a useful description and ensure the V2 Current option is selected. Click Create Key. The New API Key dialog displays.



5. Copy and save the API key. You will need this key later when defining the PagerDuty notification channel (Authorization Key). Close the dialog.
Next, you need to create a service key (Integration Key).
6. From the PagerDuty Configuration menu, select **Services**.
7. Click **Add New Service**. The Add a Service page displays.

8. Enter a name for your service and a description.
9. Under Integration Settings, select **Oracle Management Cloud** from the Integration Type drop-down menu.

 **Note:**

To narrow the list of menu options, you can enter OMC in the search field.

10. Make modifications to the Incident Settings, if necessary.
11. Click **Add Service**. The Service Details page for your newly defined service displays.

12. Copy and save the Integration Key. You will need this key later when defining the PagerDuty notification channel (Integration Key).

Now that you have both the PagerDuty Authorization (API key) and Integration keys, you define the actual PagerDuty notification channel.

13. From Oracle Management Cloud, navigate to the Notification Channels page and choose the PagerDuty notification channel type. The Create PagerDuty Channel dialog displays.

Enter the requisite information and click **Create**.

For a PagerDuty notification channel, you specify:

- **Channel Name:** An intuitive name for this notification channel.
- **Existing or New Credentials:** Choose the New Credentials option.
- **Authorization Key:** The PagerDuty API Key created earlier.
- **Integration Key:** The PagerDuty Integration Key created earlier.

Create a ServiceNow Notification Channel

You can send an Oracle Management Cloud alert to ServiceNow for incident management.

1. From the Management Cloud side menu, select **Administration**, and then **Notification Channels**.
2. On the Notification Channels page, click, **Create Notification Channel**.
3. Click on the desired notification channel type and specify the requisite information,.

Enter the ServiceNow *username*, *password*, and *instance* name as shown in the *Create ServiceNow Channel* dialog.

Create ServiceNow Channel

* Channel Name

Credentials

Existing Credentials New Credentials

* Username

* Password

Properties

* Instance

▲ Customize Payload For ServiceNow (Using Default Payload)

Use Default Payload Use Custom Payload

When creating ServiceNow incident for an alert, a default set of ServiceNow fields are automatically populated with details about the alert.
[Additional Details](#)

Create Cancel

4. Specify the payload to be sent to ServiceNow (Default Payload or Custom Payload). See below for more information about ServiceNow payloads.

 **Note:**

When specifying the ServiceNow *Instance*, do not specify the full domain. For example, if the full domain is *myinstance.service-now.com*, then you only need to enter *myinstance*.

ServiceNow Payloads

Oracle Management Cloud sends notifications to ServiceNow channels when an alert is created or updated or closed. Payload for the following three types of notifications differs from one another.

Default Payload Sent to ServiceNow

Oracle Management Cloud sends the following information by default The following table lists the default field values.

Field Name	Default Value	When it is sent:
impact	2	The alert is created.
urgency	2	The alert is created.
short_description	\${alert.message}	All updates
description	\${alert.message}	All updates
comments	<pre> \${' [code]<div style=\"padding:0 1em\"><h3>Incident Details</h3><dl style=\"padding- left:1em\"><dt>Alert ID:</dt><dd>'alert .id'</dd><dt>Event Name:</ dt><dd>'alert.eventName' </dd><dt>Event Message:</ dt><dd>'alert.message'</ dd><dt>Entity Name:</ dt><dd>'entity.name'</ dd><dt>Entity ID:</ dt><dd>'entity.id'</ dd><dt>OMC Severity:</ dt><dd>'alert.severity'< /dd><dt>Rule:</ dt><dd>'rule.ruleName'</ dd><dt>Note:</ dt><dd>'rule.note.text'< /dd><dt>Raised On:</ dt><dd>'alert.time'</ dd></dl><h4>Created by Oracle Management Cloud ServiceNow Connector</ h4></div>[/code]'} </pre>	All updates
state	6	The alert is cleared.
close_code	Solved (Work Around)	The alert is cleared.
close_notes	Management Cloud Resolution	The alert is cleared.

Customized Field Values

A field value can be a fixed string, a token, or combination of both.

When the value is a combination of fixed strings and tokens, the following rules must be followed.

- Value should start with \${

- Value should end with }
- Each fixed string should be enclosed inside a pair of single quotes.
- Token names should be entered without enclosing them inside \${ and }
- To enter consecutive multiple tokens, place a pair of single quotes between each pair of tokens for separation, e.g.,alert.id'alert.message.

Examples:

```
`${Alert ID: 'alert.id`}  
`${Alert ID: 'alert.id' Alert Message: 'alert.message`}  
`${Two tokens with a space between them: 'alert.id' 'alert.message`}  
`${Two tokens without any space between them: 'alert.id'alert.message`}
```

Create a Slack Notification Channel

You can send Oracle Management Cloud notifications to an existing Slack channel by first creating a WebHook in Slack and then creating an Oracle Management Cloud notification channel.

Create a WebHook in Slack

This channel will receive all alerts from Oracle Management Cloud.

1. Log in to your Slack account.
2. Create the incoming WebHook configuration.

`https://<youraccount>.slack.com/apps/A0F7XDUAZ-incoming-webhooks`

3. Click **Add Configuration**.
4. Select channel under *Post to Channel*.
5. Click **Add Incoming Webhooks Integration**.
6. From confirmation page, capture the WebHook URL. This URL will be required when you create the notification channel in Oracle Management Cloud.

You will need to remember the channel name as this will be used when you create the Slack notification channel in Oracle Management Cloud.

Create a Slack Notification Channel

1. From the Management Cloud side menu, select **Administration**, and then **Notification Channels**.
2. On the Notification Channels page, click, **Create Notification Channel**.
3. Click on the Slack notification channel type and specify the requisite information,.
Enter the *Channel Name* (name of the Slack channel to be displayed in the Oracle Management Cloud console), *URL* (the Slack URL), and *Team Channel* (the Slack team channel you are sending the notification to).
4. Click **Create**. A test message will be sent to Slack to confirm that the integration is working.

8

Monitor the Availability and Performance of Your Infrastructure

Monitoring your entities' health and performance is an important part of every IT administrator's job. Oracle Infrastructure Monitoring allows you to setup alerts, investigate alerts and monitor the availability status and performance of your infrastructure.

Topics

- [Typical Workflow for Monitoring the Availability and Performance of Your Infrastructure](#)
- [Monitor Availability Status](#)
- [Investigate Alerts](#)
- [Monitor Availability Status Within Each Tier](#)
- [Monitor Performance Within Each Tier](#)
- [Monitor Entity Health](#)

Typical Workflow for Monitoring the Availability and Performance of Your Infrastructure

Table 8-1 Workflow to Monitor the Availability and Performance of Your Infrastructure

Task	Description	More Information
Find out if any entities are down across the enterprise.	Identify and investigate entities that are down or have availability issues.	Monitor Availability Status
Investigate open alerts.	Review details of each open alert.	Investigate Alerts
Look for entities that are down within the tier that you manage.	Within each tier, investigate entities that are down or have availability issues.	Monitor Availability Status Within Each Tier
Identify and analyze performance issues within the tier that you manage.	Within each tier, identify entities that have potential performance problems.	Monitor Performance Within Each Tier
Check the overall health of an entity.	Check current performance of an entity.	Monitor Entity Health

Monitor Availability Status

As an administrator responsible for your entire IT infrastructure, you constantly monitor the availability status of all your infrastructure components so that you can detect and resolve problems before they affect users. Oracle Infrastructure Monitoring provides an *Entity*

Summary Dashboard that shows at a glance the current availability of all your monitored entities.

Availability Status Monitoring

- Availability status is monitored automatically
- If an entity is down, a *Down* alert of fatal severity is automatically generated. If it is a host or agent entity, a *not heard from* alert (also fatal severity) is generated.
- To get notifications for these, you must create an alert rule, choose the entity type (or entity) and choose Availability alert condition.
- Once an entity is detected to be up, the alert will clear automatically.
- If there is an error with evaluating availability status, the entity is in Error status. An alert will be generated automatically for this as well.

To monitor the current availability status across your IT infrastructure:

1. Navigate to the Enterprise Summary page and locate the Status region to view the current availability status all your entities. Note the date and time on the top-right corner of the page and make sure that you have a refreshed set of data. Set the page Auto Refresh option to a value that best matches the period during which your data needs to be refreshed.

The Entities Status section indicates the state of each entity:

- **Up** The entity is up and running, metrics are correctly collected.
 - **Error** The entity has encountered some errors, and it needs further investigation.
 - **Down** The entity is down, it isn't in a running state.
 - **Pending** The entity is in the process of being added to the monitoring service.
2. Typically, you first focus on entities that show a **Down** or **Error** status.
Drill down into the **Down** or **Error** labels and note all the entities with this status. To narrow down your list you can:
 - Filter your list of entities by type.
 - Search for a particular entity by name . For example, if you have selected entities with **Down** status, you can check one of the entity types listed on the left menu, and then search for a particular entity name to further refine your list.
 - If *global properties* are set, further filter your list by the global properties of your entities. For example, you might choose to look first into *Production* systems and later into any *non-production* systems.
 3. For every system with a **Down** or **Error** status, drill down into the *Entity Home page* for more details. Review in particular any availability alert messages on the Alerts section of the home page. Alert messages provide critical information that helps resolve availability problems. When an issue is resolved, the alert automatically clears.

To set up alert rules to send notifications for entities down or other availability issues, see [Set Up Alert Thresholds and Notifications](#).

**Note:**

You must have administrator privileges to create any alert rules.

Host-Agent Communication Monitoring

When a gateway agent cannot reach OMC, a *not heard from* alert is created for the gateway. The following alert message is generated when this occurs:

```
OMC has not received data from <gateway name> (Gateway Agent) for <N> minutes.  
It could be down or there could be network issues that impact uploading of data.  
This impacts sending status for all associated agents and its hosts, and symptom  
alerts for these will not be generated.
```

OMC will NOT generate *not heard from* alerts from the agents (and associated hosts) where the agents are communicating with OMC through the impacted gateway.

If the gateway is up later on, but the agent is still down, *not heard from* alerts will be generated on the agent and host.

Investigate Alerts

Alerts help keep your entities continuously up and running by notifying you when performance or availability problems occur.

Alerts are generated either:

1. Automatically, for all availability issues (when an entity is down or an agent is unavailable). No alert rule is required to generate these alerts.
2. Based on custom alert rules that specify a condition. For more information, see [Set Up Alert Rules](#).

Alerts indicate that a problem has occurred with one of your monitored entities. The alert details give you enough context to start investigating the problem. These details include the following:

- **Name** and **type** of entity on which the alert was raised
- Entity **Status**
- **Severity** of the problem
- **Date** and **time** when the alert was created as well as date and time of any other changes in the alert
- The **alert rule** associated with the alert which has details about the alert condition that triggered the alert and where the notification was sent

The **Alert Severity** is a key component of an alert that translates as follows:

- *Fatal*: An entity is down.
- *Critical*: A metric crosses a critical threshold.
- *Warning*: A metric crosses a warning threshold.
- *Agent Unavailable*: no recent communication has occurred between the Cloud Agent and Oracle Management Cloud. This could indicate one of the following:
 - The Cloud Agent is down.

- Even though the Cloud Agent is running, there's a connectivity problem between the Cloud Agent and Oracle Management Cloud.
- The Host on which the Cloud Agent is deployed is down.

The **Alert Details** also includes a graphical display of values of the metric being tracked and its values at various points in time. The **Alert History** keeps track of all stages of notifications.

Investigating Alerts Received

If your Oracle Infrastructure Monitoring service was set up for receiving alerts, then your administrators on duty will receive email alerts when set thresholds are exceeded or when monitored entities are down.

If your service is not yet set up for receiving alerts, see [Set Up Alert Rules](#). You must have Administrative privileges to perform this task.

Once you are setup to receive alerts, this is a typical workflow of investigating alerts that you receive:

1. Review the alert **email** and note the entity name, type, severity, and time the alert occurred. You can drill down to the alerts details directly from the email notification.
2. Click the entity name to go to its *home page*. Locate the alert in the Alerts region and click the alert message. A popup window will open, showing you the details of the alert.

You can further scroll back in recent history to find out the metric's values over time. These values should provide an indication of the problem.

3. Resolve the alert.

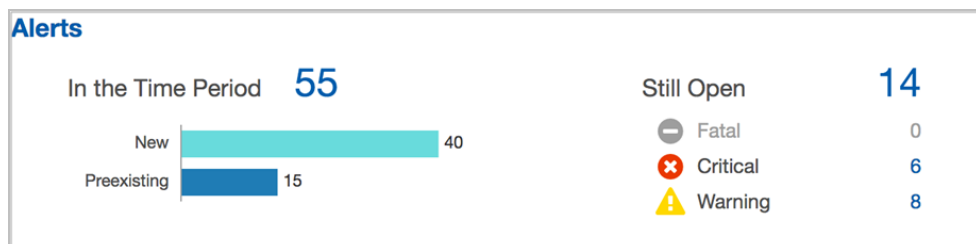
Based on your findings, make the changes required to your monitored entity and ensure that these changes won't affect other systems. When the issue is resolved, the alert will automatically clear.

Proactive Review of Alerts

Infrastructure administrators may also want to review on a daily basis the alerts triggered in the last 24 hours.

Here is a typical workflow of investigating alerts summarized on your service dashboards:

1. On the Enterprise Summary page, locate the Alerts region. The combination of up and down arrows and a number indicates an increase or decrease in alerts. If there is an increase, then drill down into that number to access the alerts page.



The Alerts are shown for a specified time period (global context). In addition to the total number of alerts in that time period, the alerts are further broken down into **New** alerts that have been raised during the period, **Preexisting** alerts that were

present at the start of the time period, as well as the number of alerts that are **Still Open** (broken down by severity).

- Investigate each newly triggered alert on the Alerts page

For any of these cases, if you determine that the alert was triggered prematurely, then consider adjusting the alerts thresholds, see [Set Up Alert Rules](#).



Note:

You must have administrator privileges to edit alert rules.

Related Alerts

Alerts for an entity can be triggered by alerts occurring on related entities. For example, a Linux host may have a WebLogic server, a Tomcat server, and multiple Oracle databases. Because these servers and databases are *related* to the host, alerts occurring on them can affect the alert status of the host itself. Specifically, related alerts are alerts that occur on related entities and that have been triggered within a 30 minute time frame (30 minutes before and 30 minutes after) the original alert.

To help you diagnose these types of related entity alert issues, you can view related alerts directly from an entity's home page.

To view related alerts:

- Navigate to an entity home page.
- From the **Alerts** tab, select an individual alert. The **Alert Details** and **Related Alerts** tabs display.
- Click on the **Related Alerts** tab as shown in the following graphic.

The screenshot shows the Oracle Alerts interface. At the top, there are navigation tabs: Alerts, Performance Charts, Performance Tables, Configuration, and Related Entities. Below the navigation, there is a section for 'Open Alerts' with a severity filter set to 'All'. A table lists several alerts with their severity (Critical, Warning, Error), time since they occurred (19 minutes 31 seconds ago), and their messages. The messages indicate issues with heap usage and memory utilization for a Tomcat server. Below the main alert list, there are two tabs: 'Alert Details' and 'Related Alerts'. The 'Related Alerts' tab is active, showing a table of alerts on related entities from July 5, 2017, 10:48:59 AM to July 5, 2017, 11:48:59 AM. The table has columns for Severity, Open Since, Alert Message, Entity Name, Entity Type, and Detail. One related alert is shown: 'Tomcat9WHAUTest(Tomcat) is Down' with Entity Name 'Tomcat9WHAUTest' and Entity Type 'Tomcat'.

Severity	Open Since	Alert Message	Service Type
Critical	19 minutes 31 seconds ago	Heap Used for [redacted] is 145 MB; it is greater than expected value of 50 MB	Monitoring
Critical	19 minutes 31 seconds ago	Heap Total for [redacted] is 192 MB; it is lower than expected value of 200 MB	Monitoring
Warning	19 minutes 31 seconds ago	Heap Free for [redacted] is 42 MB; it is lower than expected value of 100 MB	Monitoring
Critical	19 minutes 31 seconds ago	Memory Utilization for [redacted] is 97.16 %; it is greater than expected value of 50 %	Monitoring
Critical	19 minutes 31 seconds ago	Memory Usage for [redacted] is 0.1866 GB; it is greater than expected value of 0.1 GB	Monitoring

Severity	Open Since	Alert Message	Entity Name	Entity Type	Detail
Critical	Jun 27, 2017 8:20:58 AM	Tomcat9WHAUTest(Tomcat) is Down.	Tomcat9WHAUTest	Tomcat	...

“Not Heard From” Alerts on Agent and Host

- The host is monitored by the agent that is deployed on the host.
- Host availability is based on agent availability. Agent availability is based on Oracle Management Cloud receiving its performance data in regular intervals.

If there is no data received for some time, then:

- Agent and host are put in "Not Heard From" status and "Not Heard From" alerts of fatal severity are generated for the agent and host.
- To get notifications for these, create an alert rule, choose host and/or agent entities, choose availability condition and specify notification channels.
- Once the agent is back up (i.e. Oracle Management Cloud starts receiving data from the agent), then the agent and host are returned to *Up* status and the *Not Heard From* alert clears.

Monitor Availability Status Within Each Tier

For administrators responsible for various tiers of the IT infrastructure, the Oracle Infrastructure Monitoring Service Enterprise Summary dashboard provides tier regions that indicate the current status and performance of all entities in that particular tier.

The tiered status bar charts show the breakdown of status for each entity type monitored in your enterprise within that tier. For example, the following bar chart shows status of the Web Application Servers.

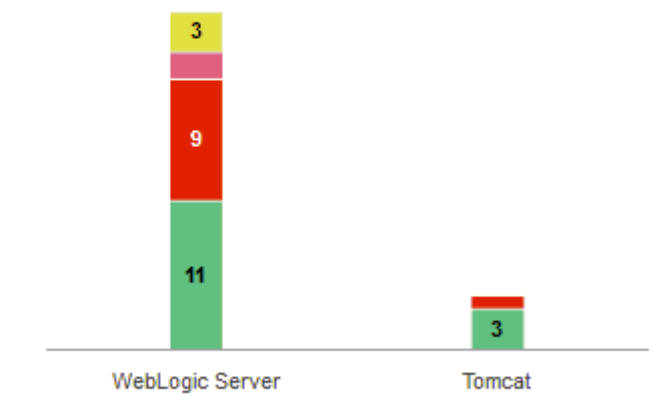
Of the total number of WebLogic Servers:

- 11 are Up (running as expected)
- 9 are Down
- 2 have Errors (more investigation needed)
- 3 are in Pending status (in the process of becoming actively monitored)

Of the total number of Tomcat servers:

- 3 are Up (running as expected)
- 1 is Down

Web Application Server



Here are some examples of tasks to perform within a tier you're investigating:

- Drill down into entities with a status other than Up.
- Identify entities related to those that don't have an Up status. For example, locate the hosts that host the Web Application Servers with a Down status.
- Review the home page for each entity that you determined is having a problem. Look for alerts and key performance metrics. Wherever applicable, entities are automatically associated and grouped as related entities. For example, application servers will automatically be associated with their corresponding database. Entities association can be viewed from the Topology display at the top of each page.

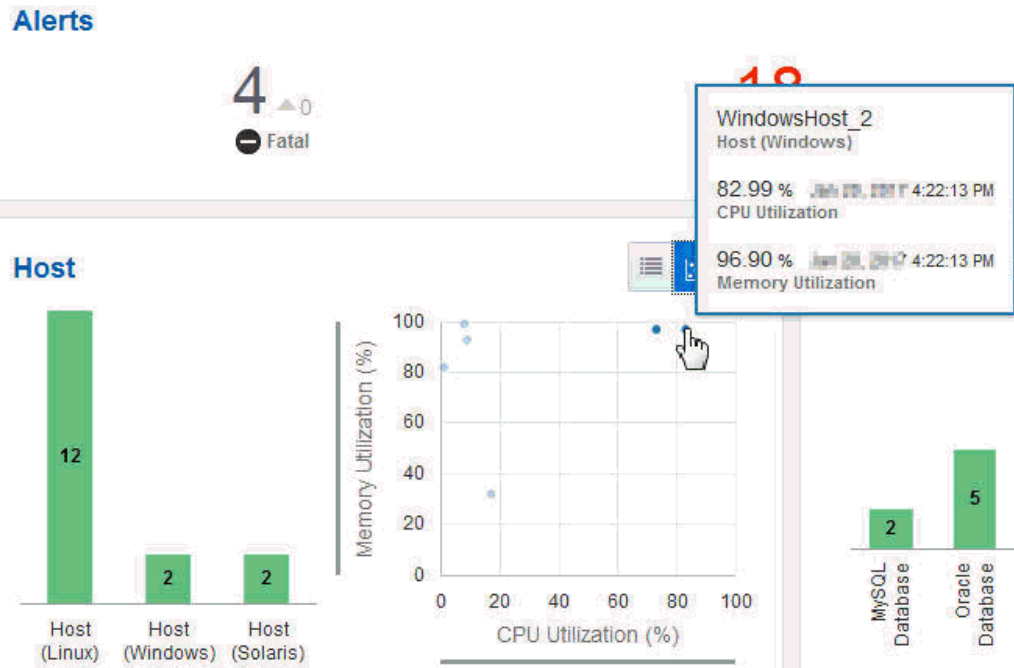
Monitor Performance Within Each Tier

For administrators responsible for various infrastructure tiers, the Enterprise Summary page provides tier regions that allow you to monitor the current performance of all entities within that tier.

The top of the Enterprise Summary page displays rolled-up information that applies to all entities: the total number of entities, the breakdown of entities status and a break-down of all alerts triggered for all entities. Below this, the graphical interface groups the entities by tiers and rolls-up status and performance information for each tier. Entities not part of any specific tier are categorized under the "Others" section. Wherever applicable, entities are automatically associated and grouped as related entities. For example, application servers will automatically be associated with their corresponding database. Entities association can be viewed from the Topology display at the top of each page.

Navigate to the Enterprise Summary page and locate the performance metrics charts for the tier you are interested in. Note first the status of all entities in your tier. Then, on the performance charts look for outliers (points on the charts that look different and are isolated compared to the others). Hover over these points to see the entity name and metric values at that point.

For example, on the CPU Load vs CPU Utilization chart, one of the points looks like an outlier. Both the CPU Utilization and Memory Utilization are high. This will require more investigation



You can further:

- Click the points on the chart to display a full history of those metrics and see if there is a trend in the metric values.
- Change the metrics displayed in the scatter chart to review the collective performance of any other two metrics. To vary the metrics displayed on each chart, select **Choose Metrics**.

Switch the performance chart to show, for example, the CPU Utilization % and Memory Utilization % across all monitored hosts. At this point you can:

- Check for outliers in this chart, look for high values of CPU Utilization % and/or Memory Utilization % which could indicate that these hosts are currently under a heavy load.
- Hover your mouse over the data point to find out which specific host is under heavy load.
- Click the data point to examine the trend of these metrics and identify how long the hosts have been under a heavy load. A long trend might indicate issues on the host that need further investigation.

While exploring your tiers, it is useful to see a sorted list of values of a particular metric, for the tier you are investigating. To help you visually assess the relative performance across entities in a tier, you can switch the display from a scatter chart to a metric table listing the top values (or bottom values) of a particular metric for all entities. This data helps to assess the most heavily loaded entities or those with the slowest performance within a tier. To correlate your findings with other related metrics, click the **Edit** button to select a new metric and assess its values for the subset of entities you are interested in.

Additional Performance Charts Controls

On the Performance charts, use the scroll wheel on the mouse to zoom in and out while maintaining the same center of the image.

You can hold down your left mouse button to select an area of data to zoom in on. When you release the mouse button, the selected area will pan to the center of the screen and automatically zoom in to fill the entire area of the chart.

The x-axis and y-axis ranges can also slide. Hold down the left mouse button and move left and right on the x-axis, or up and down on the y-axis, until you find the ideal concentration of points for your research.

Monitor Entity Health

By proactively monitoring your infrastructure, you can identify and resolve potential problems before they affect users.

The Oracle Infrastructure Monitoring **Entity Home page** enables you to proactively monitor the health of an entity. It provides an overview of all entity-related information, from entity status and open alerts to key performance indicators. Typically you reach an Entity Home page when exploring your monitored infrastructure in various ways, such as:

- Investigating a performance problem visible on the *Enterprise Summary page performance scatter charts*: Drilling down into the data point of interest allows you to reach a filtered view of the metrics in question and provides a link to the associated Entity Home page.
- Troubleshooting entities status from the *Enterprise Summary page Entity Status region*: Clicking any status provides you with a narrowed down list of all entities with that status; you can further filter your list and click the entity name to reach that Entity Home page.
- Reviewing the health and status of any entity group from the *Enterprise Summary page tiered view bar charts*: Clicking any tier provides you with a narrowed down list of all entities of that type; you can further filter your list and click on the entity name to reach that Entity Home page.
- Exploring all entities from the *Enterprise Summary page Entities region*: Drilling down into the number of entities in your infrastructure allows you to reach the Entities page where you can further filter your list and reach a particular Entity Home page.

Exploring the Entity Home page

Here is a typical set of tasks that you can perform from the Entity Home page:

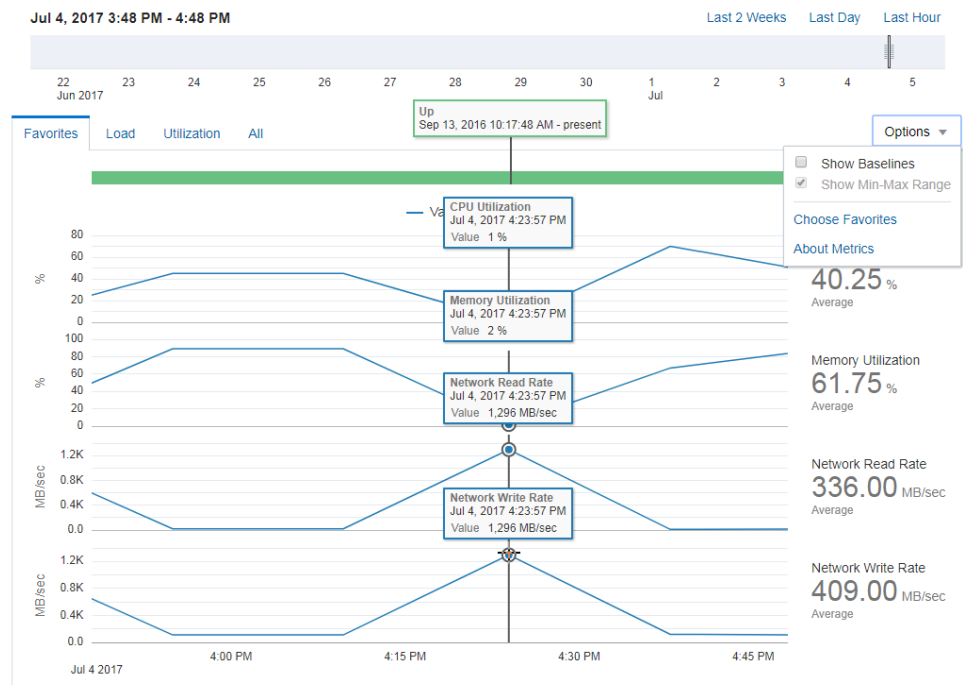
1. To reach the Entity Home page, click the name of each entity you're analyzing.

The Entity Home page has all the entity information that allows you to determine the cause of a problem. Note the following content:

- The current **availability** status displaying the entity's availability over time. Moving your cursor along the availability time line displays the corresponding time in the key performance metric charts for the entity. .
- The **open alerts** in the current time period along with their status. You can drill down on the alert numbers for more detail about those alerts.
- Key **performance metrics** for the entity. Clicking on a key performance metric displays detailed performance charts for that metric.
- The **Alerts** tab displays all alerts for the entity. You can click on an alert to view explicit details and also view related alerts (alerts generated by *related entities* that impact the currently viewed entity).

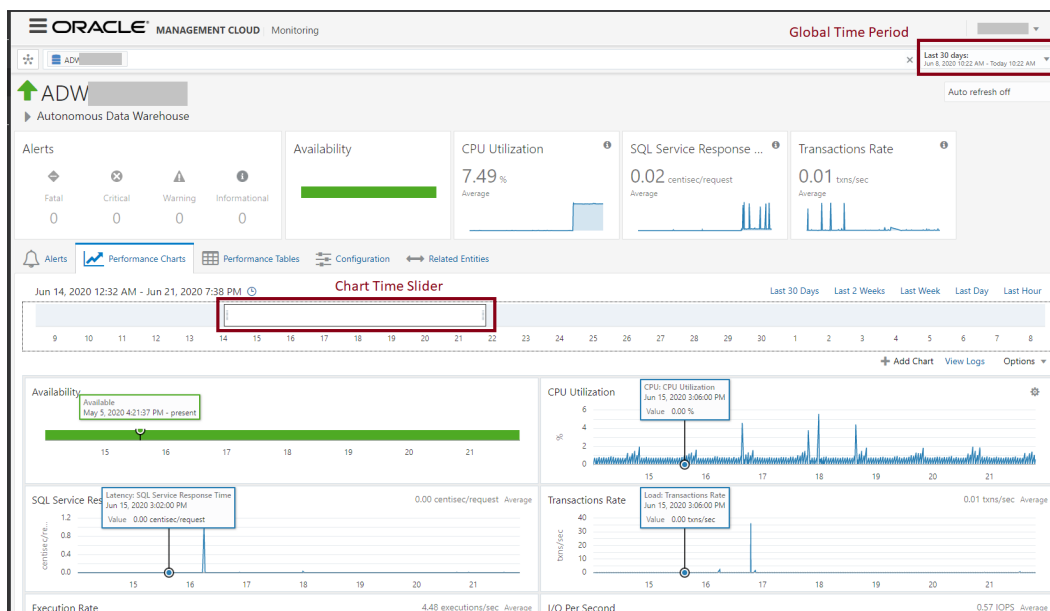
- Correlate your findings by identifying first when the entity status changed. Note the key performance metric values at that same time. Data from the last 24 hours is shown by default, and you can scroll back to more recent time periods to review the trend of the metrics over time. Hold down the left mouse button and slide the date range along the timeline until you reach the range with the data of interest. The first set of performance charts includes the key health indicators, but you can hover over any data point. You can also correlate the key performance metrics over the same time period, in all the functional categories mentioned (Capacity, Load, Response, Error, or Utilization) .

For example, you detect a performance degradation on one of your hosts. Note the key metric values at the same time.



Expanded Time Periods

In this release, Infrastructure Monitoring pages now enable you to view status and performance data up to the *Last 30 Days*. This is an enhancement over prior releases where you could view status and performance only up to the last 14 days. In the entity homepages, while you can set the Global Time Period up to the Last 30 days, in order to view performance data at its finest resolution (i.e. natively collected resolution), you'll need to set the Chart Time Slider window to at most 8 days wide.



The Chart Time Slider determines the set of data points shown in the performance charts. You can move the Slider to focus on any time period within the range specified by the Global Time Period. When the Last 30 days is chosen as the Global Time period, as long as you keep the Chart Time Slider window to show at most 8 days, you can continue to view performance data at its finest resolution up to the last 30 days. This enables you to perform better diagnostics and investigation of issues across different performance metrics.

When the Chart Time Slider window is expanded to more than 8 days, then charts will automatically switch to show the hourly rollup data. This level of control enables you to view fine-grained data when doing diagnostics or view rollup data in order to understand the trends of data across longer periods of time.

- On the performance charts, you can also select metrics to be displayed for predefined ranges, such as: Last 2 weeks, Last Day, or Last Hour. Use these preset ranges for ease of navigation. Some entities, such as relational databases, have all their properties and associated data filtered by tabs, such as: Alert, Performance Charts, Performance Tables, Configuration, and Related Entities.

Time Periods

To narrow in on a specific aspect of a monitored entity, you may only be interested in seeing a subset of metrics for that entity. For example, you have an Oracle database and you only want to see transaction volume and transaction rate displayed. You can choose which metrics you want shown and the order in which they appear by selecting **Choose Favorites** from the **Options** menu.

Baselines and Anomaly Detection

Baselines represent the normal performance of an entity that allow you to compare the current performance with previous performance and help you set appropriate thresholds for performance metrics. Baselines are calculated by observing performance metric values over a period of time and applying machine learning algorithms to this data set.

By collecting performance metrics over a period of time, Oracle Infrastructure Monitoring identifies the normal expected range of values of particular metrics and saves them as baselines. When sufficient data points are collected, daily seasonality is automatically taken into account to further fine tune baseline calculations. In this case, each metric is given an

expected range of values within each hour of a day. In addition, with more data collected, Oracle Infrastructure Monitoring also calculates the normal performance values within each day of the week. The system continues to fine-tune the data for each hour of a particular day of a week, concludes on a weekly seasonality if it is detected and includes that into the baselines calculation. For example, load metrics on a server may be expected to be at a higher range at 9:00 a.m. on a Monday and expected to be at a lower range at 9:00 a.m. on a Friday. Baselines are automatically calculated for all key performance metrics with no additional user input.

Metric values outside of the normal ranges are identified as anomalous and visually highlighted in performance charts. To receive alerts when metrics exceed normal baseline values, use the calculated baselines as guidelines and set the alert thresholds values outside of the normal ranges. For example, if a host CPU utilization is calculated to be normal between 60% and 75% on average days of the week and 65% to 80% on peak days of the week, then set your warning level alerts to 80% and a critical level alert to anything above 90%.

Metric Collection Errors

If there are errors encountered with the evaluation of a metric, then an alert of *Metric Collection Error* is generated. This alert is of critical severity.

- You can see the alerts in the Alerts UI. You can get email by creating an alert rule with the alert condition "Metric Error".
- You should look at the message of the alert and resolve the issue.
- Once the issue is resolved, this alert will clear automatically when the agent can successfully collect the metric.
- Any new metric collection errors will automatically generate an alert of 'Warning' severity instead of 'Critical' severity. All pre-existing metric collection error alerts of critical severity will remain as-is (i.e. no severity change).

9

Oracle Infrastructure Monitoring Administration Tasks

Topics

- [Typical Administration Tasks for Oracle Infrastructure Monitoring](#)
- [Maintenance Windows](#)
- [Change Monitoring Configuration](#)
- [Create and Set Global Properties](#)
- [Delete Entities](#)

Typical Administration Tasks for Oracle Infrastructure Monitoring

Table 9-1 Typical Administration Tasks for Oracle Infrastructure Monitoring

Task	Description	More Information
Change the monitoring configuration.	Update the configuration properties of an entity in Oracle Management Cloud (such as port number) to reflect the change in the entity's configuration.	Change Monitoring Configuration
Add new global properties.	Create new global properties that can be applied to existing or new entities or entity types.	Create and Set Global Properties
Set values of global properties to one or more entities.	Set predefined global properties to one or more entities.	Create and Set Global Properties
Delete monitored entities.	Delete entities you no longer want to monitor by using the <code>omcli</code> command.	Delete Entities

Maintenance Windows

A maintenance window is a period of time designated to perform regular maintenance activities on monitored entities.

A maintenance window can be scheduled to run immediately or in the future, and to run indefinitely or stop after a specific duration. Notifications, such as emails and webhooks, are suppressed during the maintenance window. However, Oracle Management Cloud continues to monitor entities and display entity status, while indicating that the entity is currently under maintenance. Although events are generated during the maintenance window, event notifications are suppressed.

For more information about maintenance windows and how to create them, see Using the Maintenance Window.

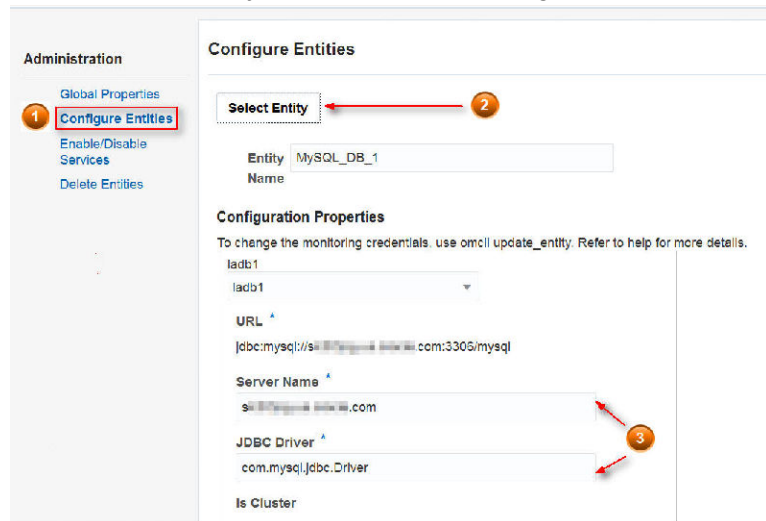
Change Monitoring Configuration

Sometimes entities need to change certain configuration parameters. Oracle Management Cloud Services must be aware of these changes.

To change the monitoring configuration of your entities, from the global menu, select the Administration option and then:

1. Select **Entities Configuration** from the global menu.
2. Select **Configure Entities** from the left menu.
3. Click on **Select Entity** to choose your entity. If you have a large number of entities, filter by Entity Type and then select your entity.
4. Change the configuration parameters and then **Save**

For example, you can change the port number your database listens on using this interface, or directory locations or other configuration attributes.



Create and Set Global Properties

You can create and set global properties in the Administration console in Oracle Management Cloud.

After you create a global property, it becomes available for all the entities present in the tenant. Setting the value of a global property for an entity allows you to group or filter the entities based on the property value.

For example, if you want to filter your Oracle Database entities by location, then you can create a global property called location, and set different values (such as Austin or Chicago) for that property for different Oracle Database entities.

Creating a Global Property

To create a global property:

1. Select **Administration** in the Management Cloud navigation menu.
2. Select **Entity Configuration** in the Administration navigation menu, then click **Global Properties**.
3. Click **Create New Global Property**.
4. In the **Create New Global Property** dialog box, enter the display name of the property and click **OK**.

The property is created and is now available to be used by an entity in the Oracle Management Cloud tenant.

Setting the Value of a Global Property

To set the value of a global property for a single or set of entities:

1. Select **Administration** in the Management Cloud navigation menu.
2. Select **Entity Configuration** in the Administration navigation menu, then click **Global Properties**.
3. In the **Select Entities** dialog box, select the entities for which you want to set the global property value and click **Select**.

The Global Properties page is displayed with the list of available global properties in the tenant.

4. In the selected entities in the **New Value** field adjacent to the property name, enter the global property value that you want to set for the selected entities, and click **Set**.

 **Note:**

If a global property is set with different values for different entities, then **(mixed)** is displayed in the **Value** column for that property.

The value for the global property is now set for the selected entities.

Delete Entities

You can delete entities that you no longer want to monitor.

Topics:

[Delete Entities from the Administration Console](#)

[Delete Entities Using omcli](#)

Delete Entities from the Administration Console

You can delete an entity from Oracle Management Cloud or view deleted entities in the Administration console.

1. Select **Administration** in the Management Cloud navigation menu.
2. Select **Entity Configuration** in the Administration navigation menu, then click **Delete Entities**.
3. In the **Select Entities** dialog box, select the entities that you want to delete and click **Select**.
The selected entities are displayed on the Delete Entities page. If you want to remove one of the selected entities, then click the **Remove** button adjacent to the entity.
4. Click **Delete <number> Entities** to delete the selected entities.

View Deleted Entities

1. On the Delete Entities page, click the **Recently Deleted** tab.
2. Select a time frame in the **View entities deleted** drop-down list.

The entities deleted within the selected time frame are displayed.

Delete Entities Using omcli

You can delete a previously added entity using the `omcli delete_entity` command as shown below.

```
<AGENT_BASE_DIR>/agent_inst/bin/omcli delete_entity agent FILENAME
```

Where *FILENAME* is the name of the file that contains the entity definition to be deleted.

Example

Say you want to delete the Oracle WebLogic Server domain that you had previously added with the following JSON file: `omc_weblogic_domain.json`.

Run the following command to delete this WebLogic Server domain:

```
<AGENT_BASE_DIR>/agent_inst/bin/omcli delete_entity agent  
omc_weblogic_domain.json
```



Note:

Cascading deletion is not supported. The JSON input for the delete command must include all entities and their sub-entities to be deleted.

10

Troubleshooting

The following sections cover common debugging procedures and diagnostic information for Infrastructure Monitoring.

Lack of Data

Issues encountered when using Infrastructure Monitoring typically fall into the following categories of lack of data in performance charts and metric collection errors.

Lack of Data in Performance Charts

For Cloud agent-monitored entities:

1. Check the time period in the UI to verify it includes the time period where data is expected.
2. Check if the entity is up. If the entity is down, no metrics can be collected.
3. Find out if the agent is up.
 - a. Go to the agent homepage.
 - b. Determine the current status of the agent via the agent command line utility. `omcli status agent`

```
$ ./omcli status agent
Oracle Management Cloud Agent
Copyright (c) 1996, 2017 Oracle Corporation. All rights reserved.
-----
Version : 1.27.0
State Home : /scratch2/agent/den00yla/lama_agent/agent_inst
Log Directory : /scratch2/agent/den00yla/lama_agent/agent_inst/
sysman/log
Binaries Location : /scratch2/agent/den00yla/lama_agent/core/1.27.0
Process ID : 85600
Parent Process ID : 90122
URL : https://myhost.myco.com:58858/emd/main/
Started at : 2018-02-21 08:26:28
Started by user : mahessub
Operating System : Linux version 3.8.13-118.20.2.el6uek.x86_64 (amd64)
Data Collector enabled : false
Sender Status : FUNCTIONAL
Gateway Upload Status : FUNCTIONAL
Last successful upload : 2018-02-23 16:04:03
Last attempted upload : 2018-02-23 16:04:03
Pending Files (MB) : 0
Pending Files : 1
Backoff Expiration : (none)
```

Agent is Running and Ready

Ensure the following:

- The agent status message displays *Agent is Running and Ready*.
 - **Last successful upload** shows a recent timestamp.
 - **Pending Files (MB)** and **Pending Files** do not show large values. If either of these statistics is large, the data may have not been uploaded yet. You should run the `omcli status agent` command periodically to verify that **Pending Files (MB)** and **Pending Files** are decreasing.
- c. Check whether the agent is able to collect the metric(s) whose data you are not seeing in the performance charts.
- i. Run the `omcli getmetric agent` command:

```
omcli getmetric agent TARGETNAME,TARGETTYPE,METRICNAME
```

 **Note:**

Make sure there are no spaces before or after comma delimiters.

If the command returns values, then it means the agent is able to collect the data.

```
$ ./omcli getmetric agent
myhost.myco.com,omc_host_linux,HOST_CPU
```

```
Oracle Management Cloud Agent
Copyright (c) 1996, 2017 Oracle Corporation. All rights
reserved.
idleTimeRaw, userTimeRaw, systemTimeRaw, waitTimeRaw, hardInterru
ptsTimeRaw, softInterruptsTimeRaw, cpuStealTimeRaw, idleTimeRawD
iff, userTimeRawDiff, systemTimeRawDiff, waitTimeRawDiff, hardInt
erruptsTimeRawDiff, softInterruptsTimeRawDiff, cpuStealTimeRawD
iff, totalTimeRawDiff, cpuUserModePercent, cpuSystemModePercent,
cpuUsageSec, cpuUtilizationPercent, cpuStolenPercent, cpuIdlePer
cent, cpuLoad1min, cpuLoad5min, cpuLoad15min, intervalSec
701388434, 5781055, 2644520, 563349, 135, 12397, 84674, 10114, 349, 50
, 3, 0, 0, 1, 10517, 3.318, 0.475, 399, 3.794, 0.01, 96.168, 0.005, 0.015,
0.0125, 26
```

- ii. Use the Agent Metric Browser to view the data the Cloud agent collects from various entities. See .
4. Check if there are metric collection error alerts on the entity home page. If there are problems collecting the metric, then no data will appear. You should see a metric collection error (Alert Type = Metric Error). To fix the metric error, ensure that all of the prerequisites for monitoring the entity type have been met.

Metric Collection Errors

Table 10-1 Debugging Metric Collection Errors

Entity Type	Metric Collection Error Message	Cause/Resolution
Oracle Database	oracle.sysman.emSDK.agent.fetchlet.exception.FetchletException: Failed to connect: java.sql.SQLRecoverableException: IO Error: The Network Adapter could not establish the connection	DB monitoring relies on a JDBC connection to the database. Check if the DB service is registered with the listener.
Oracle Pluggable Database	oracle.sysman.emd.fetchlets.db.exception.AwrFetchletException: SharedDBFetchlet - E8C3A7030E443414D10A7BAF993842BD - AWR Shared metric collection failed: java.sql.SQLRecoverableException: IO Error: The Network Adapter could not establish the connection	DB monitoring relies on a JDBC connection to the database. Ensure that the DB service is registered with the listener.
Oracle Database	oracle.sysman.emd.fetchlets.db.exception.AwrFetchletException: UDSSqlEntityFetchlet - E8C3A7030E443414D10A7BAF993842BD - AWR UDS sql entity collection failed: java.sql.SQLRecoverableException: IO Error: Socket read timed out	The root cause in most cases is a failure of the database to respond in a reasonable amount of time. This error could also be caused by network issues.
Oracle Database	orclpdb123/PDB1(Oracle Pluggable Database) is in Error state. Reason : oracle.sysman.emSDK.agent.fetchlet.exception.FetchletException: associated target does not exist or is broken apmc/ORCLPDB(Oracle Pluggable Database) is in Error state. Reason : oracle.sysman.emSDK.agent.fetchlet.exception.FetchletException: associated target does not exist or is broken	This issue occurs when the wrong JSON file has been used. Example: The CDB JSON was used to discover a PDB. For more information about Infrastructure Monitoring JSON files, see Download and Customize Oracle Infrastructure Monitoring JSONs .

Table 10-1 (Cont.) Debugging Metric Collection Errors

Entity Type	Metric Collection Error Message	Cause/Resolution
Oracle Database	Error evaluating Top SQL Executions:5MinCollection - AwrFetchletException: SharedDBFetchlet - 4A8DD99203A4341B15B61BA452994851 - internal error: java.sql.SQLException: ORA-00942: table or view does not exist	Ensure the monitoring user has sufficient privileges. For more information, see Prerequisites and Monitoring Credentials .
Oracle Database	Error evaluating CorrelationAssociations:24HrCollection-FetchletException: ORA-24247:network access denied by access control list (ACL) ORA-06512:at "SYS.UTL_INADD",line 19ORA-06512:at"SYS.UTL_INADDR",line 40 ORA-06512:at line 1	Run the PL/SQL block dbms_network_acl_admin For more information about Oracle Database prerequisites, see Prerequisites and Monitoring Credentials .
Oracle Database	Error evaluating CorrelationAssociations:24HrCollection-FetchletException: ORA24247: network access denied by access control list (ACL) ORA-06512:at"SYS.UTL_INADDR",line 19 ORA-06512: at "SYS.UTL_INADDR", Line 40 ORA-06512: at line 1	"Error evaluating SQL statistics..." The Oracle DB target needs to have DB Diagnostic Pack enabled or be an Oracle DB Enterprise Edition.
Tomcat	tomcat_mzlee (Tomcat) is in an Error state. tomcat_rest (Tomcat) is in an Error state.	
Host (Linux)	Criticaloracle.sysman.emSDK.agent.fetchlet.exception.FetchletException: Result has repeating key value : lo	

No Metric Data Appears in the UI

Steps 1 and 2 only need to be performed if the entity was newly added. If the entity was added in the past, but suddenly is not showing any metric data, complete steps 3 through 8.

1. Make sure you have completed all the prerequisite steps. See [Prerequisites and Monitoring Credentials](#).

2. Check to see if the entity type and version is supported. See [Supported Entity Types](#).
3. Go to the entity's home page and see if there are any metric collection errors.
4. Check that the entity is up.
5. Check that the agent is up.
6. Run the metrics directly using the `getmetric` command.

```
omcli getmetric agent TARGETNAME,TARGETTYPE,METRICNAME
```

To get the targetname, targettype:

```
config agent listtargets (confirm that you get the target name and type)
```

```
OPEN: metricname (this corresponds to the metricGroupName)
```

7. Check metrics in the agent metric browser.
8. Contact Oracle Support and provide debug information using the following command:

```
omcli generate_support_bundle agent DIRECTORY
```

Metrics Not Collected for Windows Environments

A Cloud agent is deployed on a Windows Server. Although the host entity has been registered in Infrastructure Monitoring performance metrics are not collected. In this situation, the root cause of the issue is performance counters in this environment were corrupt. .

To check whether performance counters are functioning properly:

1. From an administrator command prompt execute `wbemtest`. A dialog displays.
2. Click **connect**. Check the *namespace* is `root\cimv2` and click **connect** again.
3. Click **Open class** and enter the class name. For example:
`Win32_PerfRawData_PerfOS_Processor`
4. A new window will open with details about the class. If the window displays the class details, then the performance counters are working properly.
5. If any error message is displayed, execute following command as administrator from command prompt:

```
lodctr /R
```

6. After executing the above command, perform steps one through four again to verify the performance counters are working.

Create an Agent Support Bundle

When contacting Oracle Support to diagnose Oracle Management Cloud agent entity discovery issues, having comprehensive, standardized diagnostic information about your agent deployment simplifies the diagnostic process. There is a specific set of logs and

configuration information that is useful for Oracle Support to diagnose problems with the agent software. This log/configuration information is called an agent support bundle.

Generating the Agent Support Bundle

You generate the agent support bundle by running the `omcli generate_support_bundle` command. This command creates a ZIP archive containing logs and configuration information specific to your agent deployment.

Syntax

```
omcli generate_support_bundle agent DIRECTORY
```

`DIRECTORY` is the directory where you want the agent support bundle ZIP archive to be generated.

Example 10-1 Example Title

In the following example, you are generating an agent support bundle. The archive ZIP file will appear in the `/my_server/diagnostics` directory.

```
omcli generate_support_bundle agent /my_server/diagnostics
```

Host Prerequisite Validation

When installing a Cloud agent on a host, prerequisite validation checks are run. This can result in agent installation error messages shown in the following table.

Error Code	Message	Cause	Effect	Remedy
OMCAGNT-4500	OMCAGNT-4500: Host (Linux): Error occurred during prerequisite check process.	Unexpected Error during validation code execution	Validations didn't run	NA
OMCAGNT-4501	OMCAGNT-4501: This operating system [os] is not supported for host monitoring.	Is the OS supported for monitoring?	No validations for unsupported OS.	NA

Error Code	Message	Cause	Effect	Remedy
OMCAGNT-4502	<p>Host (Linux): The file [filePath] required by metric group MG cannot be either unavailable or the user agentUser doesn't have read permissions.</p> <p>Host (Solaris): The file [filePath] required by metric group MG cannot be either unavailable or the user agentUser doesn't have read permissions.</p> <p>Host (AIX): The file [filePath] required by metric group MG cannot be either unavailable or the user agentUser doesn't have read permissions.</p>	agentUser cannot access file given by path filePath .	Collection of metric group MG would be affected	Make sure agentUser can read filePath

Error Code	Message	Cause	Effect	Remedy
OMCAGNT-4503	<p>Host (Linux): The command [command] required by metric group MG is either unavailable or the user agentUser doesn't have execute permissions.</p> <p>Host (Solaris): The command [command] required by metric group MG is either unavailable or the user agentUser doesn't have execute permissions.</p> <p>Host (AIX): The command [command] required by metric group MG is either unavailable or the user agentUser doesn't have execute permissions.</p> <p>Host (Windows): The command [command] required by metric group MG is either unavailable or the user agentUser doesn't have execute permissions.</p>	<p>agentUser cannot execute command</p>	<p>Collection of metric group MG would be affected</p>	<p>Make sure command is installed on the system and agentUser can execute it</p>

Error Code	Message	Cause	Effect	Remedy
OMCAGNT-4504	<p>Host (Linux): The execution of command [command] required by metric group MG has failed due to invalid parameters.</p> <p>Host (Solaris): The execution of command [command] required by metric group MG has failed due to invalid parameters.</p> <p>Host (AIX): The execution of command [command] required by metric group MG has failed due to invalid parameters.</p> <p>Host (Windows): The execution of command [command] required by metric group MG has failed due to invalid parameters.</p>	<p>command execution failed because of invalid parameters provided</p>	<p>Collection of metric group MG would be affected (This should not happen)</p>	NA

Error Code	Message	Cause	Effect	Remedy
OMCAGNT-4505	<p>Host (Linux): The execution of command [command] required by metric group MG has timed out.</p> <p>Host (Solaris): The execution of command [command] required by metric group MG has timed out.</p> <p>Host (AIX): The execution of command [command] required by metric group MG has timed out.</p> <p>Host (Windows): The execution of command [command] required by metric group MG has timed out.</p>	<p>command validation failed because of time out. The execution didn't complete within 2mins.</p>	<p>Collection of metric group MG may be affected if the agent collection task times out.</p>	<p>NA</p>
OMCAGNT-4506	<p>Host (Windows): The service Windows Management Instrumentation (WMI) required by windows host monitoring is not in running state.</p>	<p>WMI Service is not running</p>	<p>None of the Windows Host metrics would be collected</p>	<p>Start WMI service - https://docs.microsoft.com/en-us/windows/desktop/wmisdk/starting-and-stopping-the-wmi-service</p>
OMCAGNT-4507	<p>Host (Windows): Failed to get status of service Windows Management Instrumentation (WMI) required by windows host monitoring.</p>	<p>Failing to get WMI service status</p>	<p>Validations not run.</p>	<p>NA</p>
OMCAGNT-4508	<p>Host (Windows): The WMI class [class] required by metric group MG is not available for user agentUser.</p>	<p>WMI command class class is not available for the use agentUser</p>	<p>Collection of metric group MG would be affected</p>	<p>Reload performance counters Lodctr /R More details at: https://support.microsoft.com/en-us/kb/300956</p>

Error Code	Message	Cause	Effect	Remedy
OMCAGNT-4509	Host (Windows): The check for WMI class [class] required by metric group MG has timed out.	WMI class class validation timed out. The execution didn't complete within 2 minutes.	Collection of metric group MG may be affected if the agent collection task times out.	NA

Status Unknown

If an entity shows up in Infrastructure Monitoring then it has been officially added to Oracle Management Cloud, and in turn a response has been sent back to the cloud agent (if the discovery involved a cloud agent) with the entity information.

If the entity status is shown as *Unknown*, then the possible reasons are:

- If cloud agent is involved, the agent has not sent the computed status value. In this situation, checking the agent home page and/or the agent status on the host itself may provide insight as to whether the agent is collecting the status metric for the entity. In no cloud agent is involved (entity is cloud-based), then the cloud collector has not sent the computed status value.
- The cloud agent may have a network connection issue that prevented the status value from being sent to Oracle Management Cloud. In this situation, check the network connectivity between the agent and Oracle Management Cloud.

Database Status is Shown as Down when the Database is Up

If a database entity availability status is shown as down in Oracle Management Cloud, but the database is actually up, you can perform the following troubleshooting procedures.

1. Go to the VM where the lama (Cloud) agent is located.
2. `cd $AGENT_HOME (ps -ef | grep lama)`
3. Check current status of the entities:

```
./omcli status_entity agent
```

4. `vi $AGENT_HOME/sysman/emd/targets.xml`
Search for the entity name and check `capability` property is set to `monitoring`.
5. Check the metric schedule:

```
./omcli status agent scheduler | grep Response
```

The Response metric should come against the impacted entity.

6. `./omcli getmetric agent <entity name>,omc_oracle_db,Response`
7. Check if the DB is up and check if it's able to connect using SqlPlus:

If the DB is installed in the same agent host, do the following:

```
cd $DB_HOME
export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
export ORACLE_SID=DB0221
./sqlplus "<user>/<pass>@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(Host=<host>)(Port=<port>))(CONNECT_DATA=(SID=<sid>)))"
./sqlplus "<user>/<pass>@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(Host=<host>)(Port=<port>))
(CONNECT_DATA=(SERVICE_NAME=<service_name>)))"
```

Provide the values <> with exactly the same values used while registering the DB entity.

If the DB is NOT installed in the same agent host, check if there is any DB home available in the agent home and use the DB home's SqlPlus to connect using the above steps and check the connectivity between agent and the DB.

8. Set DEBUG mode:

```
./omcli setproperty agent -allow_new -name log4j.rootCategory -
value "DEBUG, emlogAppender, emtrcAppender" -type emomslogging
```

9. First, create entity.json and then Refresh the entity:

```
./omcli refresh_entity agent <entity.json>
```

```
cat entity.json
{
  "entities": [
    {
      "name": "<entity name>",
      "type": "<entity type>"
    }
  ]
}
```

Refresh entity command can throw the error messages while connecting to the DB. This will show why the DB is shown down .

If PDB entities are shown down, but are really up:

1. PDBs have to be in READ WRITE mode in order to be shown as UP in monitoring. You can determine the PDB status from `select * from gv$containers or v$pdb`.
2. `./omcli getmetric agent <CDB entity name>,omc_oracle_db,PDB_Status`
3. Check `targets.xml` to determine whether the associations are present in both the DB entity and the PDB entity as shown below:
DB

```
<AssocTargetInstance ASSOCIATION_NAME="omc_contains"
SOURCE_TARGET_NAME="<DB entity>" SOURCE_TARGET_TYPE="omc_oracle_db"
```

```
ASSOC_TARGET_NAME="<PDB entity>" ASSOC_TARGET_TYPE="omc_oracle_pdb"/>
```

PDB

```
<AssocTargetInstance ASSOCIATION_NAME="omc_monitored_by"  
SOURCE_TARGET_NAME="<PDB entity>" SOURCE_TARGET_TYPE="omc_oracle_pdb"  
ASSOC_TARGET_NAME="<DB entity>" ASSOC_TARGET_TYPE="omc_oracle_db"/>
```


A

Monitoring Prerequisites and Credentials

This appendix contains additional prerequisite and monitoring credential information for specific entities.

Host

Prerequisites

The operating system user used to install the Cloud Agent is also used as the host monitoring credential. Your hosts are automatically added as entities when a Cloud Agent is installed. However, hosts are not automatically monitored. To enable monitoring for host entities, see [Download and Customize Oracle Infrastructure Monitoring JSONs](#).

Docker Engine / Docker Container

Docker Engine/Docker Container Configuration

You can configure a Docker Engine for monitoring in three ways:

Non-Secure Mode:

This mode doesn't need any credentials information. When the Docker Engine is configured in the non-secure mode (http), you simply need the Base URL to connect to the Docker Engine.

For example, a Base URL could be: `http://www.example.com:4243/`. Note the http, and not https mode.

To check if your Docker Engine is configured in non-secure mode, view the `/etc/sysconfig/docker` file. The following entries identify the Non-Secure Mode configuration:

```
http - non secure other_args="-H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock"
set proxy export HTTP_PROXY=<your proxy host>:80
```

You will need to provide the Docker Engine Base URL in the entity definition JSON file.

Secure Mode:

To check if your Docker Engine is configured in Secure Mode, view the `/etc/sysconfig/docker` file. If configured for:

- for 1-way SSL you will typically see an entry of the format:

```
https - secure 1 way SSL other_args="-H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock --tls --tlscert=/<certificate directory>/server-cert.pem --tlskey=/<certificate directory>/server-key.pem"
```

- for 2-way SSL you will typically see an entry of the format:

```
https - secure 2 way SSL other_args="-H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock --tlsverify --tlscacert=/<certificate directory>/ca.pem --tlscert=/<certificate directory>/server-cert.pem --tlskey=/<certificate directory>/server-key.pem"
```

If your Docker Engine is configured in Secure Mode, then you configure the monitoring credentials based on the type of communication defined.

- For **Secure 1-way SSL** you need to add the truststore *certificate* (CA certificate) in the cloud agent default truststore (*<agent home>/sysman/config/montrust/AgentTrust.jks*) using this command:

```
keytool -import -alias docker01 -keystore <agent home>/sysman/config/montrust/AgentTrust.jks -file <directory of your Docker certificate>/<certificate_file_name>.cer
```

Use the password `welcome`. Note the *<agent home>* is the directory where the Cloud Agent was installed. See *Managing Cloud Agents in Oracle® Cloud Deploying and Managing Oracle Management Cloud Agents*.

You will only need to provide the Docker Engine Base URL in the entity definition JSON file.

Docker Engine/Docker Container Configuration

- For **Secure 2-way SSL** you need to add the truststore *certificate* (CA certificate) and the *keystore* information in the agent default truststore (<agent home>/sysman/config/montrust/AgentTrust.jks).

1. Add the truststore *certificate*:

```
keytool -import -alias docker01 -keystore <agent home>/sysman/  
config/montrust/AgentTrust.jks -file <directory of your Docker  
certificate>/<certificate_file_name>.cer
```

Use the password `welcome`. Note the **agent home** is the directory where the Cloud Agent was installed.

2. Add the *keystore* information:

```
keytool -import -alias docker01 -keystore <agent home>/sysman/  
config/montrust/AgentTrust.jks -file <directory of your Docker  
certificate>/<certificate_file_name>.cer
```

Use the password `welcome`.

To add a Secure 2-way SSL Docker Engine entity you will need to create an entity definition JSON file along with a credentials JSON file. The entity definition JSON file will include your Docker Engine Base URL while the credentials file will have details about the credentials store and credentials.

For more information about how to create Docker certificates, see <https://docs.docker.com/engine/security/https/>.

Cloud Agent Configuration

If the cloud agent communicates with Oracle Management Cloud through a proxy (OMC_PROXYHOST & OMC_PROXYPORT parameters were set on the cloud agent when it was installed), Docker Engine / Docker Container discovery will fail. You'll need to perform additional configuration steps depending on the following situations:

For a New Agent Installation

If the agent requires proxy to communicate with Oracle Management Cloud, then use the gateway and set the proxy parameters (OMC_PROXYHOST & OMC_PROXYPORT) during gateway installation, and then set up the cloud agent (without proxy parameters) to point to the gateway.

For an Existing Agent

If the existing cloud agent has been set up to use the proxy to communicate with Oracle Management Cloud, to discover Docker Engine / Docker Container, execute the following commands on the cloud agent before performing entity discovery.

```
omcli setproperty agent -allow_new -name _configureProxyPerClient -  
value true  
omcli stop agent  
omcli start agent
```

XEN Virtual Platform / XEN Virtual Server

XEN Virtual Platform / XEN Virtual Server Prerequisites

To enable monitoring for XEN Virtual Platform / XEN Virtual Server, you need the root user (or) a user with SUDO privileges defined.

Oracle Database

Prerequisites

Setting Up Monitoring Credentials for Oracle Database

Before you can begin monitoring DB systems, you must have the necessary privileges. A SQL script (`grantPrivileges.sql`) is available to automate granting these privileges. This script must be run as the Oracle DB SYS user. In addition to granting privileges, the `grantPrivileges.sql` script can also be used to create new or update existing monitoring users with the necessary privileges. For information about this SQL script, location and usage instructions, see [Creating the Oracle Database monitoring credentials for Oracle Management Cloud \(Doc ID 2401597.1\)](#).

Enabling TCPS Connections

Database Side (Single Instance)

1. Create the wallets.

```
mkdir -p /scratch/aime/wallets/rwallets
mkdir -p /scratch/aime/wallets/swallets
mkdir -p /scratch/aime/wallets/cwallets
```

2. To run the `orapki` commands go to the Oracle Home and run the following commands:

```
cd $ORACLE_HOME/bin

echo "***** Create Root wallet *****"

./orapki wallet create -wallet /scratch/aime/wallets/rwallets -
auto_login -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/rwallets -dn
"C=US,O=Oracle Corporation,CN=RootCA" -keysize 2048 -self_signed -
validity 365 -pwd oracle123 -addext_ski -sign_alg sha256

./orapki wallet export -wallet /scratch/aime/wallets/rwallets -dn
"C=US,O=Oracle Corporation,CN=RootCA" -cert /scratch/aime/wallets/
rwallets/cert.pem

./orapki wallet display -wallet /scratch/aime/wallets/rwallets

openssl x509 -noout -text -in /scratch/aime/wallets/rwallets/cert.pem

echo "***** Create server wallet *****"

./orapki wallet create -wallet /scratch/aime/wallets/swallets -
auto_login -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/swallets -trusted_cert
-cert /scratch/aime/wallets/rwallets/cert.pem -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/swallets -dn
"C=US,O=Oracle Corporation,CN=DBServer" -keysize 2048 -pwd oracle123 -
addext_ski -sign_alg sha256
```

Prerequisites

```
./orapki wallet export -wallet /scratch/aime/wallets/swallets -dn
"C=US,O=Oracle Corporation,CN=DBServer" -request /scratch/aime/wallets/
swallets/csr.pem

./orapki cert create -wallet /scratch/aime/wallets/rwallets -request /
scratch/aime/wallets/swallets/csr.pem -cert /scratch/aime/wallets/
swallets/cert.pem -validity 365 -sign_alg sha256 -serial_num $(date +
%s%3N)

./orapki wallet add -wallet /scratch/aime/wallets/swallets -user_cert -
cert /scratch/aime/wallets/swallets/cert.pem -pwd oracle123

openssl x509 -noout -text -in /scratch/aime/wallets/swallets/cert.pem

./orapki wallet display -wallet /scratch/aime/wallets/swallets

echo "***** Create client wallet *****"

./orapki wallet create -wallet /scratch/aime/wallets/cwallets -
auto_login -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/cwallets -trusted_cert
-cert /scratch/aime/wallets/rwallets/cert.pem -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/cwallets -dn
"C=US,O=Oracle Corporation,CN=DBClient" -keysize 2048 -pwd oracle123 -
addext_ski -sign_alg sha256

./orapki wallet export -wallet /scratch/aime/wallets/cwallets -dn
"C=US,O=Oracle Corporation,CN=DBClient" -request /scratch/aime/wallets/
cwallets/csr.pem

./orapki cert create -wallet /scratch/aime/wallets/rwallets -request /
scratch/aime/wallets/cwallets/csr.pem -cert /scratch/aime/wallets/
cwallets/cert.pem -validity 365 -sign_alg sha256 -serial_num $(date +
%s%3N)

./orapki wallet add -wallet /scratch/aime/wallets/cwallets -user_cert -
cert /scratch/aime/wallets/cwallets/cert.pem -pwd oracle123

openssl x509 -noout -text -in /scratch/aime/wallets/cwallets/cert.pem

./orapki wallet display -wallet /scratch/aime/wallets/cwallets
```

3. Change the mode of ewallet.p12.

```
chmod 666 /scratch/aime/wallets/swallets/ewallet.p12
chmod 666 /scratch/aime/wallets/cwallets/ewallet.p12
```

Listener Changes

Prerequisites

Running SI on TCPS (Single Instance)

1. Create the Oracle Home.
2. Create a listener using TCP protocol (such as LIST).
3. Create a DB in the Oracle Home using the Listener created in Step 2. The Database and Listener might already be present.
4. Shut down the database instance.
5. Stop the Listener.

```
./lsnrctl stop LIST
```

6. Perform the following procedure.

Set the environment variables

```
export WALLET_LOCATION=/net/slc05puy/scratch/dbwallets
```

The wallet is already created and stored here. Make sure the wallet location is accessible from the current host.

```
export ORACLE_HOME=scratch/aimedb/12.1.0/12.1.0.2/dbhome_1
export ORACLE_SID=solsi
```

Back up the listener.ora, sqlnet.ora and tnsnames.ora files.

```
cp $ORACLE_HOME/network/admin/listener.ora $ORACLE_HOME/network/admin/
listener.ora.bckp
cp $ORACLE_HOME/network/admin/sqlnet.ora $ORACLE_HOME/network/admin/
sqlnet.ora.bckp
cp $ORACLE_HOME/network/admin/tnsnames.ora $ORACLE_HOME/network/admin/
tnsnames.ora.bckp
```

If sqlnet.ora is not present, create it.

```
touch $ORACLE_HOME/network/admin/sqlnet.ora
```

7. Modifying the ora files.

Listener.ora

Replace all 'TCP' with 'TCPS'

```
sed -i 's/TCP/TCPS/' $ORACLE_HOME/network/admin/listener.ora
```

Replace all '43434' with '2484' [43434 being the old listener port number]

```
sed -i 's/34343/2484/' $ORACLE_HOME/network/admin/listener.ora
```

Before executing the above shell commands, make sure you don't have any

Prerequisites

string other than the protocol which contains "TCP". This also applies to the for Listener port.

```
echo "SSL_CLIENT_AUTHENTICATION = TRUE" >> $ORACLE_HOME/network/admin/
listener.ora;
```

```
echo "WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY
= $WALLET_LOCATION/swallets)))" >> $ORACLE_HOME/network/admin/
listener.ora;
```

```
echo "SSL_VERSION = 1.2" >> $ORACLE_HOME/network/admin/listener.ora;  **
Only if TLS version has to be 1.2
```

```
[SSL_VERSION = 1.2 or 1.1 or 1.0]
```

```
Sqlnet.ora
```

```
echo "SQLNET.AUTHENTICATION_SERVICES= (BEQ, TCPS)" >> $ORACLE_HOME/
network/admin/sqlnet.ora;
```

```
echo "SSL_CLIENT_AUTHENTICATION = TRUE" >> $ORACLE_HOME/network/admin/
sqlnet.ora;
```

```
echo "WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY
= $WALLET_LOCATION/swallets)))" >> $ORACLE_HOME/network/admin/sqlnet.ora;
```

```
echo "SSL_VERSION = 1.2" >> $ORACLE_HOME/network/admin/sqlnet.ora;
** Only if TLS version has to be 1.2
```

8. Start the listener (`./lsnrctl start LIST`)
9. Start the database instance.
10. Run `./lsnrctl status LIST` and check if the listener is running on TCPS with 2484 as the port and is associated with the database.

```
./lsnrctl status LTLS
LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 06-APR-2016
13:03:54
Copyright (c) 1991, 2014, Oracle. All rights reserved.
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=myhost.myco.com)(PORT=2484)))
```

```
STATUS of the LISTENER
```

```
-----
Alias                LTLS
Version              TNSLSNR for Linux: Version 12.1.0.2.0 -
Production
Start Date           06-APR-2016 10:41:33
Uptime               0 days 2 hr. 22 min. 21 sec
Trace Level          off
Security             ON: Local OS Authentication
SNMP                 OFF
```

Prerequisites

Listener Parameter File
/scratch/12102tls12/product/dbhome_1/network/admin/listener.ora

Listener Log File
/scratch/12102tls12/diag/tnslsnr/myhost/1tls/alert/log.xml

Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=myhost.myco.com)
(PORT=2484)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC2484)))

Services Summary...
Service "sitls" has 1 instance(s).
Instance "sitls", status READY, has 1 handler(s) for this service...
Service "sitlsXDB" has 1 instance(s).
Instance "sitls", status READY, has 1 handler(s) for this service...
The command completed successfully.

You can see in the example that the database is now associated with the listener. If it is not, check whether the database `local_listener` parameter is set to the listener's connect descriptor.

```
alter system set local_listener='<CONNECT_DESCRIPTOR FOR NEW LISTENER
PORT>';
```

Example: `alter system set local_listener='(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS) (HOST=strka31.myco.com) (PORT=2484)))';`

Once done, bounce the database instance. Even after doing this, if the database is not getting associated with the listener, and the listener is up and running without any issue, go to the ORACLE HOME and create a brand new database out of it using DBCA. It will prompt you to use the listener you just secured, and which is up and running on TCPS protocol.

TCPS Credentials

In order to establish secure communication with the Oracle Database, you must add TCPS Database Credential Properties to the credential JSON file in order to add the Oracle Database entity.

- **connectionTrustStoreLocation:** Your server/trust Key Store Location. This property is used to specify the location of the trust store. A trust store is a key store that is used when making decisions about which clients and servers can be trusted. The property takes a String value that specifies a valid trust store location.
- **connectionTrustStoreType:** Your server/trust Key Store Type. This property denotes the type of the trust store. It takes a String value. Any valid trust store type supported by SSL can be assigned to this property.
- **connectionTrustStorePassword:** Your server/trust Key Store Password. This property is used to set the password for the trust store. The trust store password is used to check the integrity of the data in the trust store before accessing it. The property takes a String value.
- **connectionKeyStoreLocation:** Your client Key Store Location. This property is used to specify the location of the key store. A key store is a database of key material that are used for various purposes, including authentication and data integrity. This property takes a String value.
- **connectionKeyStoreType:** Your client Key Store Type. This property denotes the type of the key store. It takes a String value. Any valid key store type supported by SSL can be assigned to this property.
- **connectionKeyStorePassword:** Your client Key Store Password. This property specifies the password of the key store. This password value is used to check the integrity of the data in the key store before accessing it. This property takes a String value.

Prerequisites

Agent Properties**Client authority**

```
./omcli setproperty agent -name connectionKeyStoreLocation -value /scratch/  
aime/wallets/cwallets/ewallet.p12  
./omcli setproperty agent -name connectionKeyStoreType -value sha256  
./omcli setproperty agent -name connectionKeyStorePassword -value oracle123
```

Server authority

```
./omcli setproperty agent -name connectionTrustStoreLocation -value /  
scratch/aime/wallets/swallets/ewallet.p12  
./omcli setproperty agent -name connectionTrustStorePassword -value  
oracle123  
./omcli setproperty agent -name connectionTrustStoreType -value sha256
```

Once set, bounce the Agent.

```
./omcli stop agent  
./omcli start agent
```

 **Note:**

Make sure that the above wallet is accessible at the agent location.

AWS-RDS Oracle DB

Prerequisites

See [Monitor AWS - RDS Oracle DB](#).

Oracle Automatic Storage Management (ASM)

Credentials

Monitoring of ASM is supported through credential-based monitoring. For simplicity, use the default `asmnmp` user for the ASM monitoring credentials OR any user with both SYSASM and SYSDBA roles.

 **Note:**

For monitoring ASM, the agent should be version 1.47 or above.

Oracle NoSQL

Credentials

Monitoring of Oracle NoSQL is supported only through credential-less JMX (no credentials JSON file is needed).

MySQL Database

Prerequisites

To enable monitoring for a My SQL Database, you can create a special database user, for example, `moncs` as follows:

1. Create a user:

```
CREATE USER 'moncs'@'l hostname' IDENTIFIED BY 'password';
```

2. Grant appropriate privileges:

```
GRANT SELECT, SHOW DATABASES ON *.* TO 'moncs'@'hostname' IDENTIFIED  
BY 'password';  
GRANT SELECT, SHOW DATABASES ON *.* TO 'moncs'@'%' IDENTIFIED BY  
'password';
```

3. Flush privileges.
-

Microsoft SQL Server

Prerequisites

To enable monitoring for a Microsoft SQL Server Database, you can create a special database user as follows.

Create a user (for example, `moncs`) and map the new user to the `master` and `msdb` databases. Then, give this user the following minimum privileges.

 **Note:**

Beginning with Oracle Management Cloud 1.31, `sqladmin`-related privileges are no longer required.

```
CREATE LOGIN moncs
WITH PASSWORD = 'moncs';
GO
CREATE USER moncs FOR LOGIN moncs;
GO
```

Then, map the user `moncs`:

1. From the **Security** menu, select **Logins** `moncs`.
2. Right-click on `moncs` and select **Properties**.
3. Select **User Mapping**.
4. Map to all system and user databases:

```
USE master;
GRANT VIEW ANY DATABASE TO moncs;
GRANT VIEW ANY definition to moncs;
GRANT VIEW server state to moncs;
GRANT SELECT ON [sys].[sysaltfiles] TO [moncs];
GRANT execute on sp_helplogins to moncs;
GRANT execute on sp_readErrorLog to moncs;

GRANT EXECUTE ON dbo.xp_regread TO moncs;
```

```
USE msdb;
GRANT SELECT on dbo.sysjobsteps TO moncs;
GRANT SELECT on dbo.sysjobs TO moncs;
GRANT SELECT on dbo.sysjobhistory TO moncs;
```

For connecting to SQL server database with SSL encryption, do the following:

1. Ensure the SQL server installation has the required updates for TLS 1.2 support as described in the following document.
<https://support.microsoft.com/en-in/help/3135244/tls-1-2-support-for-microsoft-sql-server>

Prerequisites

2. Create a server certificate for the SQL server host.
Set up the certificate as mentioned in the section “Install a certificate on a server with Microsoft Management Console (MMC)” in the following document: <https://support.microsoft.com/en-in/help/316898/how-to-enable-ssl-encryption-for-an-instance-of-sql-server-by-using-mi>
3. Install the server certificate for the SQL server instance.
Set up the SQL server instance to use the server certificate created above, as mentioned in the section “To install a certificate for a single SQL Server instance” in the following document: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/manage-certificates?view=sql-server-2016>
4. Export to a file, the root certification authority’s certificate that has signed the SQL server host certificate, and copy this file to the cloud agent host.
Export the certificate as described in section “Enable encryption for a specific client” in the following document: <https://support.microsoft.com/en-in/help/316898/how-to-enable-ssl-encryption-for-an-instance-of-sql-server-by-using-mi>
5. Create a trust store on the cloud agent host, and import the root certification authority’s certificate exported above.

```
keytool -import -file .\ca_cert.cer -alias mytrust -
keystore .\trustStore.jks -storetype jks
```

6. Form the connection URL pointing to the trust store.

```
jdbc:sqlserver://
xxx.xxx.com:1433;encrypt=true;trustServerCertificate=false;trustSto
re=C:\trustStore.jks;trustStorePassword=xxxx;
```

MongoDB Database

Prerequisites

To enable monitoring for a MongoDB Database, you can create a special database user, for example, `omc_monitor` as follows:

1. Connect to your database:

```
use your MongoDB database name;
```

2. Create user:

```
db.createUser(
{
user: "omc_monitor",
pwd: "mongo123",
roles: [ "read" ]
}
)
```

Oracle WebLogic Server (includes WebLogic Domain and WebLogic Cluster)

Prerequisites

To enable monitoring of a Oracle WebLogic Server (WLS), use a WebLogic user with at least the Monitor role. The user can also have Operator or Administrator roles, which include the Monitor role.

If you have enabled the Oracle WebLogic Server with SSL, you must export the certificate from its keystore and import it in the Cloud Agent keystore. Perform the following steps:

1. Stop the Cloud Agent.

```
omcli stop agent
```

2. Export the certificate from the WLS instance JMX SSL keystore to the Cloud Agent's truststore. For example, on a UNIX host:

```
cd <agent base Directory>/agentStateDir/sysman/config/montrust  
keytool -exportcert -alias <alias of WLS SSL key> -file <Exported  
Cert Name> -keystore <path to the WLS SSL Keystore>.keystore -  
storepass <WLS SSL Keystore password> -rfc
```

3. Import the WLS instance JMX SSL keystore to the Cloud Agent's truststore:

```
keytool -import -noprompt -alias <alias agent's truststore key> -file  
<Exported Cert Name>.cer -keystore AgentTrust.jks -storepass <Agent  
truststore password, default is "welcome">
```

4. Restart the Cloud Agent..

```
omcli start agent
```

Oracle Service Bus

Prerequisites

Important: Before you can monitor Oracle Service Bus (OSB) entities in Oracle Management Cloud, you must first enable monitoring from the Oracle Service Bus Administration console.

ORACLE[®] Service Bus 11gR1

The screenshot displays the Oracle Service Bus Administration console interface. On the left, the 'Change Center' shows a 'weblogic session' with options to 'Activate', 'Discard', or 'Exit'. Below it, the 'Project Explorer' shows a tree view of projects including 'default', 'J2eeApp', 'BusinessServices', 'ProxyServices', 'v2', 'v3', and 'wsdls'. The main area shows the configuration for a 'Proxy Service (J2eeApp/ProxyServices/v2/ CustomerPS)'. A table lists metadata: 'Last Modified By' (weblogic), 'Last Modified On' (1/21/19 2:42 AM), 'References' (2 Ref(s)), and 'Referenced By' (0). Below the table are tabs for 'Configuration Details', 'Operational Settings', 'SLA Alert Rules', and 'Policies'. The 'General Configuration' section shows 'State' as 'Enabled'. The 'Monitoring' section has 'Enable Pipeline Monitoring' checked, with a dropdown menu set to 'Action' level or above. 'Aggregation Interval' is set to 0 hours and 1 minute. 'SLA Alerts' are enabled at 'Normal' level or above.

Property	Value	Description
Last Modified By	weblogic	
Last Modified On	1/21/19 2:42 AM	
References	2 Ref(s)	
Referenced By	0	

General Configuration

State	<input checked="" type="checkbox"/> Enabled
-------	---

Monitoring

Monitoring	<input checked="" type="checkbox"/> Enable Pipeline Monitoring at Action level or above
Aggregation Interval	0 hours 1 mins
SLA Alerts	<input checked="" type="checkbox"/> Enable Alerting at Normal level or above

For information, see [What are Operational Settings for a Service?](#).

Once monitoring has been enabled from the Oracle Service Bus Administration console, you can add OSB entities to Oracle Management Cloud. When specifying an OSB entity, you use credentials of a user with at least the **Monitor** role. The user can also have either the **Operator** or **Admin** role.

Tomcat

Prerequisites and Credentials

Tomcat is monitored using JMX. You must configure Tomcat for JMX remote monitoring even if you are using a local agent.

Tomcat can be monitored with or without authentication. If a JMX credential is created, then it's assumed you're monitoring this entity with credentials.

To create a JMX credential for monitoring:

1. Edit the environment file:

```
vi $CATALINA_HOME/bin/setenv.sh
```

Add:

```
CATALINA_OPTS="-Dcom.sun.management.jmxremote -  
Dcom.sun.management.jmxremote.port=9999 -  
Dcom.sun.management.jmxremote.ssl=false -  
Dcom.sun.management.jmxremote.authenticate=true -  
Dcom.sun.management.jmxremote.password.file=./conf/  
jmxremote.password -Dcom.sun.management.jmxremote.access.file=./  
conf/jmxremote.access"
```

2. Save the file.
3. Change the file permission as executable:

```
chmod 755 $CATALINA_HOME/bin/setenv.sh
```

4. Edit the password file:

```
vi $CATALINA_HOME/conf/jmxremote.password
```

Add:

```
control tomcat  
admin tomcat
```

5. Edit the access file:

```
vi $CATALINA_HOME/conf/jmxremote.access
```

Add:

```
control readonly  
admin readwrite
```

Prerequisites and Credentials

6. Change the file permission for only the owner:

```
chmod 600 jmxremote.access
chmod 600 jmxremote.password
```

7. Bounce the Tomcat instance:

```
sh $CATALINA_HOME/bin/shutdown.sh
sh $CATALINA_HOME/bin/startup.sh
```

If you have enabled the Tomcat JMX with SSL, you must export the certificate from its keystore and import it in the Cloud Agent keystore. Perform the following steps:

1. Export the certificate from the Tomcat instance JMX SSL keystore to the Cloud Agent's truststore. For example, on a UNIX host:

```
cd <agent Base Directory>/agentStateDir/sysman/config/montrust
keytool -exportcert -alias <alias of Tomcat JMX SSL key> -file
<Exported Cert Name>.cer -keystore <path to the Tomcat JMX SSL
Keystore>.keystore -storepass <Tomcat JMX SSL Keystore password> -rfc
```

2. Import the Tomcat instance JMX SSL keystore to the Cloud Agent's truststore:

```
keytool -import -noprompt -alias <alias agent's truststore key> -file
<Exported Cert Name>.cer -keystore AgentTrust.jks -storepass <agent
truststore password, default is "welcome">
```

3. Restart the agent, using the command line interface:

```
omcli stop agent
omcli start agent
```

Oracle Traffic Director (OTD)

Prerequisites

OTD 11

Use an OTD Administrator user.

In addition, to enable collection of metrics, you must configure and start an SNMP subagent. To start the SNMP subagent, use OTD Admin Console, or use the following command:

```
tadm start-snmp-subagent
--host=<otd_host>
--port=<otd_port>
--user=<otd user>
--password-file=<password_file>
```

For more information on configuring and starting an SNMP subagent, see the Oracle Traffic Director documentation.

OTD 12

Use a WebLogic Server user with the Monitor role. The user can also have Operator or Admin roles, which include the Monitor role.

Apache HTTP Server

Apache HTTP Server Prerequisites

In this release, only Apache HTTP Server 2.4.x and 2.2 for Linux are supported.

To enable the collection of configuration metrics, note the following:

1. The Cloud Agent should be installed on the same host as Apache HTTP Server. The Apache *.conf file(s) , including httpd.conf file, should be accessible and readable by the Cloud Agent install user.
2. The Apache install user and the Cloud Agent install user should be a part of the same operating system group.

In order to monitor an Apache HTTP Server you must first:

- Enable 'mod_status' for the Apache module.
- Configure/server-status location directive for the specified Host and Port (default or configured virtual host).
- Turn 'ON' the Extended Status.
- If applicable, provide access to the configured location directive so that HTTP/HTTPS request can be successfully made from the host where the agent is installed on.

For more information, see https://httpd.apache.org/docs/2.4/mod/mod_status.html and <http://httpd.apache.org/docs/current/mod/core.html#location>.

For HTTPS/Secure communication between Apache HTTP Server and the cloud agent during metrics collection, you must provide an SSL certificate. To make the certificate available with the cloud agent:

1. Append the contents of your certificate file to the existing certificate file. For example, on a UNIX host the existing certificate file is: <AGENT_BASE_DIR>/sysman/config/b64InternetCertificate.txt

Ensure that only the following lines are appended to the b64InternetCertificate.txt file. Do not include blank lines, comments, or any other special characters.

```
-----BEGIN CERTIFICATE-----  
<<<Certificate in Base64 format>>>  
-----END CERTIFICATE-----
```

2. Restart the agent by running the following commands from the agent installation directory (for example, on a UNIX host, this directory is <AGENT_BASE_DIR>/agent_inst/bin).

```
a) ./omcli stop agent  
b) ./omcli start agent
```

For data retrieval of memory-related metrics (supported on Unix platforms and when an entity is locally monitored), the PID file (httpd.pid) file needs to be accessed.

If Apache is running as root or some user other than the agent process owner, access to the PID file will fail. Hence, to allow access to httpd.pid, you need to ensure that the file can be accessed without compromising Linux security. There are several ways to achieve this. One option is as follows:

As a privileged user, run the following commands:

```
setfacl -R -L -d -m u:<agent_user>:rx /etc/httpd/run  
setfacl -R -L -m u:<agent_user>:rx /etc/httpd/run
```

Apache HTTP Server Prerequisites

where `/etc/httpd/run` is the directory containing the PID file.


Oracle HTTP Server (OHS)

Prerequisites

OHS 12 : Node Manager credentials are required.

Also, the following prerequisites must be met:

- Cred-less (No credential file to be provided when running `omcli add_entity` during discovery) OHS discovery when the standalone OHS process owner and agent process owner are same user.
- Cred-based: OHS discovery when the standalone OHS process owner and agent process owners are different users.

 **Note:**

cred-less and cred-based discovery is applicable for standalone OHS 11. For OHS 12, only cred-based discovery is supported

- For HTTPS/Secured communication between OHS and the Cloud agent (for metric data collection) , the required certificate must be available with the agent in order for the SSL handshake to be successful. To make the certificate available with the agent :
 - Append the contents of your certificate file to the file : `/sysman/config/b64InternetCertificate.txt`
 - Ensure that only the following lines are appended to the `b64InternetCertificate.txt` file (that is, do not include blank lines, comments, or any other special characters):

```
-----BEGIN CERTIFICATE-----
<<<Certificate in Base64 format>>>
-----END CERTIFICATE-----
```

- Restart the agent by running the following commands :

```
omcli stop agent;omcli start agent;
```

Arista Ethernet Switch

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string (which was entered during the Arista Switch configuration) along with IP address of agent that will be used for Arista Switch monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus the authentication method (SHA or MD5) and authorization password if authorization is used. In addition, you must supply the privilege method (only DES is supported) and privilege password if privilege is used. Everything needs to be manually configured up front in the Arista Switch.

Read-only access is all that's required for Arista Switch monitoring.

Cisco Ethernet (Catalyst) Switch

Prerequisites

To enable monitoring of the Cisco Ethernet (Catalyst) Switch, you will need to provide the SNMPv1/v2 or SNMPv3 credentials in the JSON credential file. Read-only access is sufficient for Cisco Catalyst Switch monitoring. For more information on how to configure an SNMP user for a Cisco Catalyst Switch, see http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960/software/release/12-2_55_se/configuration/guide/scg_2960/swsnmp.html#78160

Cisco Nexus Ethernet Switch

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string (which was entered during Cisco Nexus Ethernet Switch configuration) along with IP address of agent that will be used for Cisco Nexus Ethernet Switch monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus the authentication method (SHA or MD5) and authentication password if authentication is used,. In addition, the privilege method (only DES supported) and privilege password must be supplied if privilege is used. Everything needs to be manually configured up front in the Cisco Nexus Ethernet Switch.

Read only access is enough for the Cisco Nexus Ethernet Switch monitoring.

Oracle Power Distribution Unit (PDU)

Prerequisites

To enable monitoring, HTTP and SNMPv1/v2c/v3 are needed. The NMS and trap tables in PDU administration interface must be set for a proper SNMP monitoring. for more information, see the PDU vendor documentation.

Juniper Ethernet Switch

Prerequisites

To enable monitoring, HTTP and SNMPv1/v2c/v3 are needed.

If SNMPv1/v2 is used, you must provide SNMP community string that has been used earlier in Juniper Switch configuration along with IP address of agent which will be used for Juniper Switch monitoring.

If SNMPv3 is used, in addition to SNMPv3 user, you must provide the auth method (SHA or MD5) and auth-password if auth is used, and priv method (only DES supported) and priv-password if priv used. You must configure everything manually in Juniper Switch. Read only access is sufficient for Juniper Switch monitoring.

Oracle Infiniband Switch

Prerequisites

To enable monitoring, HTTP and SNMPv1/v2c/v3 are needed.

If SNMPv1/v2 is used, you must provide SNMP community string that has been used earlier in IB Switch configuration along with IP address of agent which will be used for IB Switch monitoring.

If SNMPv3 is used, in addition to SNMPv3 user, you must provide the auth method (SHA or MD5) and auth-password if auth used, and plus priv method (only DES supported) and priv-password if priv used. You must configure everything manually in IB Switch. Read only access is sufficient for IB Switch monitoring.

Brocade Fibre Channel Switch

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string (entered during Brocade Fibre Channel Switch configuration), along with the IP address of the agent that will be used for Brocade Fibre Channel Switch monitoring.

If SNMPv3 is used, you must provide the SNMPv3 the user, plus the authentication method (SHA or MD5) and authorization password (if authorization is used), plus privilege method (only DES is supported) and privilege password if a privilege method is used. All of this needs to be manually configured up front in the Brocade Fibre Channel Switch.

Read-only access is enough for Brocade Fibre Channel Switch monitoring.

SCOM (System Center Operations Manager)

Prerequisites

Credentials must follow the same criteria as any program which tries to obtain data from SCOM using the SCOM SDK. See [How to Connect an Operations Manager SDK Client to the System Center Data Access Service](#).

... The account that is used for authentication must be included in an Operations Manager user-role profile ...

The OMC Cloud Agent uses the *omc_scom.exe* client to connect to the SCOM SDK. The Cloud agent does not bundle required SCOM SDK libraries (due to the license type of libraries). You must manually copy the SCOM SDK libraries to the machine where the agent is running.

```
C:\Program Files\Microsoft System Center 2012 R2\Operations
Manager\Server\SDK Binaries\Microsoft.EnterpriseManagement.Runtime.dll
C:\Program Files\Microsoft System Center 2012 R2\Operations
Manager\Server\SDK
Binaries\Microsoft.EnterpriseManagement.OperationsManager.dll
C:\Program Files\Microsoft System Center 2012 R2\Operations
Manager\Server\SDK Binaries\Microsoft.EnterpriseManagement.Core.dll
```

Juniper SRX Firewall

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must supply the SNMP community string (which was entered during Juniper SRX Firewall configuration) along with IP address of agent that will be used to monitor the Juniper SRX Firewall.

If SNMPv3 is used, you must supply the SNMPv3 user, plus the authentication method (SHA or MD5) and authentication password, if authentication is used. In addition, privilege method (only DES supported) and privilege password will be required, if privileges are used. Everything must be manually configured up front in the Juniper SRX Firewall.

Read-only access is sufficient for Juniper SRX Firewall monitoring.

Fujitsu Server

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during XSCF configuration along with IP address of the agent that will be used to monitor the Fujitsu Server.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authorization method (SHA or MD5) and authorization password if authorization used. You must also provide the privilege method (only DES supported) and privilege password if privileges are used. All of this must be manually configured upfront in the Fujitsu server service processor..

Read-only access is adequate for the monitoring.

For more information on how to configure SNMP users in Fujitsu M10 servers, see <http://www.fujitsu.com/downloads/SPARCS/manuals/en/c120-e684-11en.pdf>

Intel/SPARC Computers

Credentials

Only the username and password are required to use SSH to log in to the ILOM service processor.

VMware vCenter

Prerequisites

In order for the Cloud Agent to be able to collect all the metrics for the Oracle Management Cloud VMware entities, you should:

1. Install VMware tools on the VM host.
2. Set the statistics level to one (1).

Credentials: username/password required to access VMware vCenter (use Administrator role).

Example:

```
username=Administrator@vsphere.local / password=<admin_pw>
```

Certificates:

You need to explicitly add the vCenter certificate to the Agent's JKS:

Example:

```
<jdk>/bin/keytool -importcert -file <vmware-vsphere-certificate> -alias vmware  
-keystore $T_WORK/agentStateDir/sysman/config/montrust/AgentTrust.jks -  
storepass welcome
```

How to extract certificate from vCenter:

```
openssl s_client -showcerts -connect <hostname>:443
```

Discovery properties:

How to retrieve VMware vCenter Server Instance UUID to be passed in at discovery time through the entity property *omc_virtual_mgmt_system_id* using VMware PowerCLI:

Example:

```
PS C:\> $vcenter = Connect-viserver vcsa-01a.corp.local -User  
Administrator@vsphere.local -Password admin_pw  
PS C:\> $vcenter.InstanceUuid  
d322b019-58d4-4d6f-9f8b-d28695a716c0
```

Docker Swarm

Prerequisites and Credentials

Cloud Agent Configuration

If the cloud agent communicates with Oracle Management Cloud through a proxy (OMC_PROXYHOST & OMC_PROXYPORT parameters were set on the cloud agent when it was installed), Docker Swarm discovery will fail. You'll need to perform additional configuration steps depending on the following situations:

For a New Agent Installation

If the agent requires proxy to communicate with Oracle Management Cloud, then use the gateway and set the proxy parameters (OMC_PROXYHOST & OMC_PROXYPORT) during gateway installation, and then set up the cloud agent (without proxy parameters) to point to the gateway.

For an Existing Agent

If the existing cloud agent has been set up to use the proxy to communicate with Oracle Management Cloud, to discover Docker Swarm, execute the following commands on the cloud agent before performing entity discovery.

```
omcli setproperty agent -allow_new -name _configureProxyPerClient -  
value true  
omcli stop agent  
omcli start agent
```

Credentials

There are three methods you can use to authenticate and connect to the Docker Swarm via Rest APIs

- 1) Non-secure
 - 2) Secure (https): 1-way SSL mode
 - 3) Secure (https): 2-way SSL mode
-

Apache SOLR

Prerequisites

Two modes are supported: standalone & solrcloud

Monitoring is done over REST APIs exposed by Apache SOLR

Monitoring credentials require read access to following URIs:

- /admin/collections?action=clusterstatus
- /admin/collections?action=overseerstatus
- /admin/info/system
- /admin/info/threads
- /admin/cores
- /<core_name>/admin/mbeans

Credentials

1. Without Credentials:
 - a. non-secure (http)
 - b. secure (https)
2. With Credentials:
 - a. Client Authentication - (2-way SSL)
 - b. Basic Authentication - non secure
 - c. Basic Authentication - secure
 - d. Basic Authentication with Client authentication

Hadoop Cluster

Prerequisites

By default, Hadoop runs in non-secure mode in which no actual authentication is required.

By configuring Hadoop to run in secure mode, each user and service needs to be authenticated by Kerberos in order to use Hadoop services.

To perform Kerberos authentication, the Cloud Agent requires the following:

1. krb5.conf file. This file can be found at /etc/krb5.conf
2. Username and password

The Cloud Agent can use only one krb5.conf at a time. If a single Agent needs to perform Kerberos authentication with more than one domain, these details should be defined in a single krb5.conf file.

Arbor TMS/CP

Prerequisites

SNMP v1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during Arbor appliance configuration along with IP address of Cloud Agent which will be used for appliance monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the appliance..

Read-only access is adequate for Arbor appliance monitoring.

Juniper Netscreen Firewall

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during firewall configuration along with IP address of the Cloud Agent which will be used for Juniper firewall monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the firewall..

Read-only access is adequate for Juniper firewall monitoring.

Juniper MX Router

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during router configuration along with the IP address of the Cloud Agent which will be used for router monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the router.

Read-only access is adequate for MX router monitoring.

F5 BIG-IP LTM

Credentials

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during F5 BIG-IP LTM configuration along with IP address of Cloud Agent which will be used for the LTM monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the LTM.

Read-only access is adequate for LTM monitoring.

F5 BIG-IP DNS

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during F5 BIG-IP DNS configuration along with IP address of the Cloud Agent which will be used for the DNS monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the DNS.

Read-only access is adequate for DNS monitoring.

ES2 Ethernet Switch

Prerequisites

SNMPv1/v2 or SNMPv3 credentials needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during ES2 configuration along with IP address of the Cloud Agent which will be used for appliance monitoring.

If SNMPv3 used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) password if authentication is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the appliance.

Read-only access is adequate for the ES2 monitoring.

Oracle Flash Storage

Prerequisites and Credentials

Oracle Flash Storage exposes monitoring data through the REST API.

Oracle Flash Storage credentials (username and password) are required to monitor Oracle Flash Storage.

Apache Cassandra DB

Prerequisites

The default settings for Cassandra make JMX accessible only from the local host. If you want to enable remote JMX connections, change the `LOCAL_JMX` setting in `cassandra-env.sh` and enable authentication and/or SSL. To do this, perform the following procedure:

1. Open the `cassandra-env.sh` file for editing and update or add these lines:

```
JVM_OPTS="$JVM_OPTS -  
Dcom.sun.management.jmxremote.authenticate=true"  
JVM_OPTS="$JVM_OPTS -  
Dcom.sun.management.jmxremote.password.file=/etc/cassandra/  
jmxremote.password"
```

If the `LOCAL_JMX` setting is in your file, set it to **no**:

```
LOCAL_JMX=no
```

2. Depending on whether the JDK or JRE is installed:
 - Copy the `jmxremote.password.template` from `/jdk_install_location/jre/lib/management/` to `/etc/cassandra/` and rename it to `jmxremote.password`

```
$ cp /jdk_install_dir/lib/management/  
jmxremote.password.template /etc/cassandra/jmxremote.password
```
 - Copy the `jmxremote.password.template` from `/jre_install_location/lib/management/` to `/etc/cassandra/` and rename it to `jmxremote.password`

```
$ cp /jre_install_dir/lib/management/  
jmxremote.password.template /etc/cassandra/jmxremote.password
```
3. Change the ownership of the `jmxremote.password` to the user you use to run Cassandra and change permission to read-only:

```
$ chown cassandra:cassandra /etc/cassandra/jmxremote.password  
$ chmod 400 /etc/cassandra/jmxremote.password
```

4. Edit `jmxremote.password` and add the user and password for JMX-compliant utilities:

```
monitorRole QED  
controlRole R&D  
cassandra cassandrapassword
```

Note:

The Cassandra user and Cassandra password shown in the above sample are examples. Specify the user and password for your environment.

Prerequisites

5. Add the Cassandra user with read and write permission to /jre_install_location/lib/management/jmxremote.access

```
monitorRole readonly
cassandra readwrite
controlRole readwrite \
create javax.management.monitor.,javax.management.timer. \
unregister
```

6. Restart Cassandra.
-

Oracle VM Server for SPARC (LDoms)

Prerequisites

- Prerequisites: OMC Cloud Agent is deployed on the LDoms Control Domain
- Discovery does not require any user credentials but you need to grant *solaris ldoms read RBAC* privileges to the OMC Cloud Agent user:

```
/usr/sbin/usermod -A solaris.ldoms.read oracle
```

- Discovery properties:
 - The following command retrieves the LDoms Control Domain UUID to be supplied at discovery time through entity identifying property *omc_virtual_platform_id* using *virtinfo*:

```
# virtinfo -ap | grep DOMAINUUID
DOMAINUUID|uuid=280c9ff4-a134-48cd-cee9-a270b2aaefa0
```

- Autodiscovery of LDoms-related entities:
 - Use a JSON file with details to discover the Oracle VM Server for SPARC (LDoms). Using this method, all Logical Domains (Virtual Machines) are automatically discovered and updated periodically when things change in the Oracle VM Server for SPARC (LDoms) deployment.
-

Coherence

Prerequisites

Supports both credential and non-credential monitoring. When using a secured JMX connection, a credential input file needs to be passed. For information on configuring a Coherence cluster, see [Configure a Coherence Cluster](#).

Oracle Unified Directory(ODU)

Prerequisites

- i) OUD Gateway
- ii) OUD Replication
- iii) OUD Proxy

LDAP username and LDAP passwords are used to connect to the OUD LDAP server.

OUD Credentials:

Directory Server Username and Password: The username and password that will be used by the agent to bind to the server instance. Ensure the password is in the appropriate field.

The following credential JSON sample illustrates how the properties should be entered.

```
{ "entities":[
  {
    "name":"OMC_OUD_Directory1",
    "type":"omc_oud_directory",
    "displayName":"OUD_directory1",
    "timezoneRegion":"PST",
    "credentialRefs":["OudCreds"],
    "properties":{
      "host_name":{"displayName":"Directory Server
Host","value":"myserver.myco.com"},
      "omc_ldap_port":{"displayName":"Administration
Port","value":"4444"},
      "omc_trust_all":{"displayName":"Trust ALL Server SSL
certificates","value":"true"},
      "capability":
{"displayName":"capability","value":"monitoring"}}
    }
  ]
}

{"credentials":[
  {
    "id":"OudCreds","name":"OUD
Credentials","credType":"MonitorCreds",
    "properties":[{"name":"authUser", "value":"CLEAR[cn=Directory
Manager]"},
      {"name":"authPasswd",
"value":"CLEAR[mypassword]"}]
    }
  ]
}
```

Oracle Access Manager (OAM)

Prerequisites and Monitoring Credentials

The same credentials are used to discover the WebLogic Domain.

 **Note:**

Refresh of IDM targets is now supported. To refresh any IDM domain run `omcli refresh_entity agent ./idm_domain.json` where the content of `idm_domain.json` is:

```
{ "entities":[
  {
    "name": "Idm Domain",
    "type": "omc_weblogic_domain"
  }
]}
```

Oracle Internet Directory (OID)

Prerequisites

Same credentials used to discover the WebLogic Domain.

 **Note:**

Refresh of IDM targets is now supported. To refresh any IDM domain run `omcli refresh_entity agent ./idm_domain.json` where the content of `idm_domain.json` is:

```
{ "entities":[
  {
    "name": "Idm Domain",
    "type": "omc_weblogic_domain"
  }
]}
```

Microsoft Internet Information Services (IIS)

Prerequisites

Local Monitoring: Credentials are not required. The agent user is used for monitoring.

Remote Monitoring via WMI: Credentials are required. The credentials to be provided include the username and password used to log into the remote Windows host.

Before you can monitor Microsoft IIS entities, you must ensure the following prerequisites have been met:

- *Remote Monitoring of IIS:* If the Cloud agent and IIS are installed on different machines, then Microsoft Visual C++ needs to be installed on the Windows machine running the Cloud agent. The DLL *msvcr100.dll*, which is part of the Microsoft Visual C++ installation, is required.

Local Monitoring of IIS: If the Cloud agent and IIS are installed on the same machine, Microsoft Visual C++ is not required.

- IIS has been installed on a Windows Server. For more information about running the installation wizards from Server Manager, see [Installing IIS 8.5 on Windows Server 2012 R2](#).
- IIS Management Compatibility Components have been installed. To install the components:
 1. Click **Start**, click **Control Panel**, click **Programs and Features**, and then click **Turn Windows features on or off**.
 2. Follow the installation wizards and on the **Select Server Roles** page, select **Web Server (IIS)**. For more information about running the installation wizards from Server Manager, see [Installing IIS 8.5 on Windows Server 2012 R2](#).
 3. In Server Manager, expand Roles in the navigation pane and right-click Web Server (IIS), and then select Add Role Services.
 4. In the Select Role Services pane, scroll down to Web Server>Management Tools. Check the following boxes:
 - IIS 6 Management Compatibility
 - IIS 6 Metabase Compatibility
 - IIS 6 Management Console
 - IIS 6 WMI Compatibility
 5. Enable FTP Server.
- DCOM settings and WMI namespace security settings have been enabled for a remote WMI connection.

WMI uses DCOM to handle remote calls. DCOM settings for WMI can be configured using the DCOM Config utility (**DCOMCnfg.exe**) found in **Administrative Tools** in **Control Panel**. This utility exposes the settings that enable certain users to connect to the computer remotely through DCOM.

The following procedure describes how to grant DCOM remote startup and activation permissions for certain users and groups.

1. Click **Start**, click **Run**, type **DCOMCNFG**, and then click **OK**
2. In the **Component Services** dialog box, expand **Component Services**, expand **Computers**, and then right-click **My Computer** and click **Properties**
3. In the **My Computer Properties** dialog box, click the **COM Security** tab
4. Under **Launch and Activation Permissions**, click **Edit Limits**
5. In the **Launch Permission** dialog box, follow these steps if your name or your group does not appear in the **Groups or user names list**

Prerequisites

- a. In the **Launch Permission** dialog box, click **Add**
- b. In the **Select Users, Computers, or Groups** dialog box, add your name and the group in the **Enter the object names to select** box, and then click **OK**
6. In the **Launch Permission** dialog box, select your user and group in the **Group or user names** box. In the **Allow** column under **Permissions for User**, select **Remote Launch** and select **Remote Activation**, and then click **OK**

The following procedure describes how to grant DCOM remote access permissions for certain users and groups.

1. Click **Start**, click **Run**, type **DCOMCNFG**, and then click **OK**
2. In the **Component Services** dialog box, expand **Component Services**, expand **Computers**, and then right-click **My Computer** and click **Properties**
3. In the **My Computer Properties** dialog box, click the **COM Security** tab
4. Under **Access Permissions**, click **Edit Limits**
5. In the **Access Permission** dialog box, select **ANONYMOUS LOGON** name in the **Group or user names** box. In the **Allow** column under **Permissions for User**, select **Remote Access**, and then click **OK**

Allowing Users Access to a Specific WMI Namespace

It is possible to allow or disallow users access to a specific WMI namespace by setting the "Remote Enable" permission in the WMI Control for a namespace.

The following procedure sets remote enable permissions for a non-administrator user:

1. In the **Control Panel**, double-click **Administrative Tools**
2. In the **Administrative Tools** window, double-click **Computer Management**
3. In the **Computer Management** window, expand the **Services and Applications** tree and double-click the **WMI Control**
4. Right-click the **WMI Control** icon and select **Properties**
5. In the **Security** tab, select the namespace and click **Security**
6. Locate the appropriate account and check **Remote Enable** in the **Permissions** list

Firewall Settings

1. Click **Start**, click **Run**, type **GPEDIT.MSC**, and then click **OK**
 2. In the **Group Policy** dialog box, expand **Administrative Templates**, expand **Network**, expand **Network Connections**, and then expand **Windows Firewall**
 3. Select **Standard Profile** and double click on Windows Firewall : Allow Inbound Remote Administration Exceptions
 4. In the dialogue box that pops up, select **Enabled** and click on **Apply**
 5. If required, repeat the above 2 steps for **Domain Profile** as well
-

Oracle Identity Manager (OIM)

Prerequisites

Same credentials used to discover the WebLogic Domain.

Note:

Refresh of IDM targets is now supported. To refresh any IDM domain run `omcli refresh_entity idm_domain.json` where the content of `idm_domain.json` is:

```
{ "entities":[
  {
    "name": "Idm Domain",
    "type": "omc_weblogic_domain"
  }
]}
```

Oracle Clusterware (CRS)

Prerequisite for Remote Monitoring

:

SSH must be set up between the machine where the Cloud agent is installed and the machine where CRS is installed, The Cloud agent connects to the remote machine where CRS is installed via SSH authentication.

JBOSS


Prerequisites

Before discovering a JBOSS server or domain, you must first add the JBOSS client jar file to the Cloud agent as a plug-in. The JBOSS client jar file contains the required JMX protocols that allow the agent to collect JBOSS metrics.

The JBOSS client jar is distributed as part of the JBOSS installation. When you download the JBOSS zip file, the client jar file will be bundled with it.

Step	Action
Step 1: Locate the JBOSS client jar file.	<p>From the JBOSS home directory, you will find the client jar file at the following location:</p> <pre>> JBOSS_HOME/bin/client</pre> <p>In this directory, you'll see the <code>jboss-client.jar</code> file. This is the file you need to copy over to the Cloud agent location.</p>

Step	Action
Step 2: Copy the JBOSS client jar file to the Cloud agent installation.	Copy the <i>jboss-client.jar</i> file to a secure location that is accessible by the Cloud agent. Typically, this is located on the same host where the agent is installed.
Step 3: Add the <i>jboss-client.jar</i> to the Cloud agent installation as a plug-in.	<p>From the Cloud agent home directory, navigate to the agent state directory:</p> <pre data-bbox="876 451 1242 493"><agent_home>/sysman/config</pre> <p>Create a classpath file. This file tells the agent where to find the <i>jboss-client.jar</i>. The file naming convention is <i><plugin_id>.classpath.lst</i>.</p> <p>Example: If you're adding the GFM plug-in (plug-in ID is <i>oracle.em.sgm</i>), the file name would be <i>oracle.em.sgm.classpath.lst</i>.</p> <p>Edit the classpath file and add the absolute path to the <i>jboss-client.jar</i> file at the end of the file.</p> <pre data-bbox="876 808 1291 871">/scratch/securelocation/jboss-client.jar</pre> <p>Bounce the agent. Any modifications made to the classpath file will not take effect until the agent is restarted. Once the agent has been bounced, you are ready to discover the JBOSS entity (server or domain).</p>

Step	Action
Step 4: Discover the JBOSS server/domain.	<ol style="list-style-type: none"> 1. From the Oracle Management Cloud console, select AdministrationàDiscoveryàAdd Entity. The Add Entity page displays. 2. From the Entity Type drop-down menu, choose either JBOSS Domain or JBOSS Server. The appropriate JBOSS parameters are displayed. 3. Enter the appropriate parameters and monitoring credentials. 4. Click Add Entity. <p>About JBOSS Monitoring Credentials</p> <p>Depending on whether you choose JBOSS Server or JBOSS Domain entity type, the required monitoring credentials will differ:</p> <p><i>JBOSS Server</i></p> <ul style="list-style-type: none"> • JBOSS Username: User account used by the agent for monitoring. • JBOSS Password: Password for the above user account. <p><i>JBOSS Domain</i></p> <ul style="list-style-type: none"> • JBOSS Credentials: <ul style="list-style-type: none"> – JBOSS username and password: Credentials used by the agent for monitoring. – App User Name and Password: Credentials used to communicate with servers in the domain. • User Credential Set: <ul style="list-style-type: none"> – Alias and Password: Same as the JBOSS username and password used for the JBOSS Credentials. <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>This is needed because two different fetchlets are being used.</p> </div>

Kubernetes Cluster

In order to monitor Kubernetes, you need to set access permissions.

Cloud Agent Configuration

If the cloud agent communicates with Oracle Management Cloud through a proxy (OMC_PROXYHOST & OMC_PROXYPORT parameters were set on the cloud agent when it was installed), Kubernetes discovery will fail. You'll need to perform additional configuration steps depending on the following situations:

For a New Agent Installation

If the agent requires proxy to communicate with Oracle Management Cloud, then use the gateway and set the proxy parameters (OMC_PROXYHOST & OMC_PROXYPORT) during gateway installation, and then set up the cloud agent (without proxy parameters) to point to the gateway.

For an Existing Agent

If the existing cloud agent has been set up to use the proxy to communicate with Oracle Management Cloud, to discover Kubernetes, execute the following commands on the cloud agent before performing entity discovery.

```
omcli setproperty agent -allow_new -name _configureProxyPerClient -value true
omcli stop agent
omcli start agent
```

Master's Server Certificate

To connect to the Kubernetes API server, you need to add the Master's Server Certificate to the agent trust store.

Downloading the Certificate (Command Line)

```
# echo -n | openssl s_client -connect <master host>:<master port> | sed -ne
'/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > <certificate name>.cert
# Example for Kubernetes Master URL https://myhost.myco.com:6443/ execute
the following
$ echo -n | openssl s_client -connect myhost.myco.com:6443 | sed -ne '/-
BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > "slcak155.crt"
```

Adding to the Agent Trust Store

For secure Kubernetes installations (https), it is mandatory to add the certificate of the Kubernetes master to the agent's trust store. This can be done either using OMCLI or during the discovery process. If you add the certificate to the agent trust store using `omcli secure`, then you don't need to fill out the certificate-related fields (Certificate,CertAlias,CertPassword) in the JSON or the UI when adding the target. Alternatively, you can add the certificate to the agent trust store during the discovery process (either through the UI or using `omcli add_entity`).

- Using OMCLI

```
# omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc
<path to the certificate> -alias <alias of the certificate>
$ omcli secure add_trust_cert_to_jks -password welcome trust_certs_loc /
agentdir/cloud_agent/agent_inst/bin/slcak155.crt -alias kube-server
```

- During Discovery

From the UI (Token Credentials, Basic Credentials, Keystore Credentials)

ORACLE MANAGEMENT CLOUD

Add Entity

Before proceeding further, make sure that all the [prerequisites](#) are met. If your entity type is not on the list, refer to the [documentation](#) for adding entities using omcli.

* Entity Type:

Discover Using Credentials:

* Entity Name:

* Kubernetes Master URL:

Host Name:

Heapster URL:

* Cloud Agent:

Monitoring Credentials

These are the credentials that will be used by the cloud agent for monitoring. The credentials will not be saved in Oracle Management Cloud.

Token Credentials Basic Credentials Keystore Credentials

* Token:

Keystore Certificate:

Certificate Alias:

Trust Store Password: [Show](#)

From the Command Line

When running `omcli add_entity`, the path to the certificate, its alias, and the agent's trust store password can be specified as part of the credential JSON, as shown in the following JSON example (Token-based authentication).

```
{
  "credentials": [
    {
      "id": "KubeCredsRef",
      "name": "TokenCredentialSet",
      "credType": "TokenCredential",
      "properties": [
        {
          "name": "Token",
          "value": "CLEAR[<Token>]"
        },
        {
          "name": "Certificate",
          "value": "FILE[<Absolute path of Kubernetes certificate
file>]"
        },
        {
          "name": "CertAlias",
          "value": "CLEAR[<Kubernetes certificate alias>]"
        },
        {
          "name": "CertPassword",
          "value": "CLEAR[<Agent trust store password>]"
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

Token Based Authentication

Creating a Service Account

You can reuse a service account already present in the Kubernetes installation or create a new one.

```

# kubectl create serviceaccount <service account name>
$ kubectl create serviceaccount omc-monitoring

```

To list available service accounts:

```

$ kubectl get serviceaccounts
NAME                SECRETS  AGE
default             1        14d
omc-monitoring      1        1h

```

Getting a Token

Every service account when created will have a secret associated with it.

```

$ kubectl get secrets
NAME                DATA      AGE      TYPE
default-token-ggjlh 3          14d     kubernetes.io/service-account-token
omc-monitoring-token-96jpc 3          1h     kubernetes.io/service-account-token
# We are interested in the name that starts with our serviceaccount name.
Here for example the token for omc-monitoring serviceaccount is omc-
monitoring-token-96jpc

```

The token value can be extracted by describing the secret.

```

# kubectl describe secrets <secret name>
$ kubectl describe secrets omc-monitoring-token-96jpc
Name:          omc-monitoring-token-96jpc
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name=omc-monitoring
              kubernetes.io/service-account.uid=belff6c9-ed72-11e8-
b9e7-0800275fc834
Type:         kubernetes.io/service-account-token
Data
====
ca.crt:      1025 bytes
namespace:   7 bytes
token:       eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9... (extracted token)

```

Access Permissions

Before the token of the service account is used for monitoring we need the "get, list and watch" privileges for following resources:

Kubernetes Version	Resources
1.8	pods,nodes,namespaces,services,jobs,services/ proxy,deployments.apps,replicasets.apps,daemonsets.apps,statefulsets.apps,replicationcontrollers
1.7	pods,nodes,namespaces,services,jobs,services/ proxy,deployments.apps,replicasets.extensions,daemonsets.extensions,statefulsets.apps,replicationcontrollers
1.9 to 1.11	pods,nodes,namespaces,services,jobs,services/ proxy,deployments.apps,replicasets.apps,daemonsets.apps,statefulsets.apps,replicationcontrollers,nodes/proxy
1.5 and 1.6	pods,nodes,namespaces,services,jobs,services/ proxy,deployments.extensions,replicasets.extensions,daemonsets.extensions,statefulsets.apps,replicationcontrollers

For adding the permissions to a service account, first create a cluster role

```
# kubectl create clusterrole <cluster_role_name> --verb=get,list,watch
--resource=<Resources>
$ kubectl create clusterrole omc-monitoring-role --
verb=get,list,watch --
resource=pods,nodes,namespaces,services,jobs,services/
proxy,deployments.apps,replicasets.apps,daemonsets.apps,statefulsets.ap
ps,replicationcontrollers,nodes/proxy
```

After the cluster role is created, it must be bound to the service account.

```
# kubectl create clusterrolebinding <cluster_role_binding_name> --
clusterrole=<cluster_role_name> --
serviceaccount=default:<service_account_name>
$ kubectl create clusterrolebinding omc-monitoring-binding --
clusterrole=omc-monitoring-role --serviceaccount=default:omc-
monitoring
```

Note:

If you want access to all the resources you can use this clusterrole **cluster-admin**, which has all privileges and can create binding for the created serviceaccount using below command.

```
# kubectl create clusterrolebinding <cluster_role_binding_name> --
clusterrole=cluster-admin --
serviceaccount=default:<service_account_name>
```



```
$ kubectl create clusterrolebinding all-access-binding --clusterrole=cluster-admin --serviceaccount=default:all-access-account
```

Basic Authentication

Creating Authorization Policy

For a specific user, you must create a username and password and add them in a file "**basic_auth.csv**" (under directory **letc/kubernetes/pki** on the master node) as follows:

```
<password>,<username>,<groupname>
```

In a new file "**abac-authz-policy.jsonl**" (under directory **letc/kubernetes/pki** on the master node). For above user, you need to specify what API's the user needs access to and with what privileges, as shown below.

```
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "pods",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "nodes",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "services",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "namespaces",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "deployments",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "daemonsets",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "replicasets",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "jobs",  
"apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource":  
"replicationcontrollers", "apiGroup": "*", "readonly": true}}  
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind": "Policy",  
"spec": {"user": "<username>", "namespace": "*", "resource": "statefulsets",  
"apiGroup": "*", "readonly": true}}
```

 **Note:**

If you need to provide access to all resources for this user, just add the following command in the JSON file.

```
{"apiVersion": "abac.authorization.kubernetes.io/v1beta1", "kind":  
"Policy", "spec": {"user": "<username>", "namespace": "*", "resource":  
"*", "apiGroup": "*", "readOnly": true}}
```

Modifying the API Server Manifest

Add details of above two files in `/etc/kubernetes/manifests/kube-apiserver.yaml` on the master node as follows.

- `--authorization-mode=Node,RBAC,ABAC` (EDIT THIS ENTRY)
- `--basic-auth-file=/etc/kubernetes/pki/basic_auth.csv` (NEW ENTRY)
- `--authorization-policy-file=/etc/kubernetes/pki/abac-authz-policy.jsonl` (NEW ENTRY)

 **Note:**

1) Make sure that while adding the above lines, use spaces for indentation (do not use tabs). 2) Also, if you are trying to change the policy file, you need to copy them in a new file and refer the new file name as above, since APIServer will not be able to identify if there is any change in the existing policy JSON file.

To reflect the changes, restart the kubelet using the following command:

```
# Run the following command as root user  
$ systemctl restart kubelet
```

Client Certificate Authentication

Add following lines to `/etc/kubernetes/manifests/kube-apiserver.yaml`

- `--client-ca-file=/srv/kubernetes/ca.crt`
- `--tls-cert-file=/srv/kubernetes/server.crt`
- `--tls-private-key-file=/srv/kubernetes/server.key`

Information on how to generate these certificates can be found at the following website:

<https://kubernetes.io/docs/concepts/cluster-administration/certificates/>

To create the JKS from certificates, run the following:

```
cat server.crt server.key > keyCert.pem
openssl pkcs12 -export -in keyCert.pem -out clientKeystore.pkcs12 -name
clientKeystore -noiter -nomaciter
keytool -importkeystore -srckeystore clientKeystore.pkcs12 -srcstoretype
pkcs12 -srcaalias clientKeystore -destkeystore clientKeystore.jks -
deststoretype jks -deststorepass <password> -destalias clientKeystore
```

Discovery

Kubernetes can be discovered from either the Oracle Management Cloud console or from OMCLI.

OMCLI

Following table shows the content of the sample JSON files used to discover Kubernetes using OMCLI. See [Add Entities Using JSON Files](#).

Without credentials - insecure

```
{
  "entities": [
    {
      "name":
      "KUBERNETES_INSECURE",
      "type":
      "omc_kubernetes_cluster",
      "displayName":
      "KUBERNETES_INSECURE",
      "timezoneRegion": "GMT",
      "properties": {
        "host_name": {
          "displayName":
          "Hostname",
          "value":
          "myhost.myco.com"
        },
        "omc_kubernetes_master_url": {
          "displayName":
          "Kubernetes master URL",
          "value": "http://
myhost.myco.com:80/"
        },
        "capability": {
          "displayName":
          "capability",
          "value": "monitoring"
        }
      }
    }
  ]
}
```

With credentials - secure

```
{
  "entities": [
    {
      "name": "KUBERNETES_SECURE",
      "type":
      "omc_kubernetes_cluster",
      "displayName":
      "KUBERNETES_SECURE",
      "credentialRefs": [
        "KubeCredsRef"
      ],
      "timezoneRegion": "GMT",
      "properties": {
        "host_name": {
          "displayName":
          "Hostname",
          "value":
          "myhost.myco.com"
        },
        "omc_kubernetes_master_url": {
          "displayName":
          "Kubernetes master URL",
          "value": "https://
myhost.myco.com:443/"
        },
        "capability": {
          "displayName":
          "capability",
          "value": "monitoring"
        }
      }
    }
  ]
}
```

 **Note:**

An additional property `omc_heapster_url` can be specified in the JSON's "properties" object (shown below) in order to fetch metrics from heapster. If this property is not provided then the metrics will be fetched from summary API.

```
{
  .....
  "properties" : {
    ....
    "omc_heapster_url" : {
      "displayName" : "Heapster URL",
      "value" : "<base url of heapster>"
    }
  }
}
```

Property description

Property	UI Name	Description
<code>omc_kubernetes_master_url</code>	Kubernetes Master URL	Base URL of the API Server on the Kubernetes Master Node. The URL is of the form <code>http(s)://<hostname>:<port></code>
<code>host_name</code>	Hostname	Hostname of the Kubernetes master node
<code>omc_heapster_url</code>	Heapster URL	Base URL of Heapster. This needs to be specified if the performance metrics are to be collected from Heapster. If heapster is running inside Kubernetes as a cluster service the Base URL is of the form <code>http(s)://<host>:<port>/api/v1/namespaces/kube-system/services/heapster/proxy</code> Here the host & port are same as in <code>omc_kubernetes_master_url</code>

Credential JSONs

Basic Credentials	Token Credentials	Keystore Credentials
<pre>{ "credentials": [{ "id": "KubeCredsRef", "name": "UserCredentialSet", "credType": "AliasCredential", "properties": [{ "name": "Alias", "value": "CLEAR[admin]" }, { "name": "Password", "value": "CLEAR[M3ASfn0poA4tMdc O]" }], "name": "Certificate", "value": "FILE[<KUBERNETES_CERT _FILE_LOC>]" }, { "name": "CertAlias", "value": "CLEAR[<KUBERNETES_CER T_ALIAS>]" }, { "name": "CertAlias", "value": "CLEAR[<KUBERNETES_CER T_ALIAS>]" }] }</pre>	<pre>{ "credentials": [{ "id": "KubeCredsRef", "name": "TokenCredentialSet", "credType": "TokenCredential", "properties": [{ "name": "Token", "value": "CLEAR[seRsr3jMfQL81Dq vSgqgjJwH65j80gzB]" }, { "name": "Certificate", "value": "FILE[<KUBERNETES_CERT _FILE_LOC>]" }, { "name": "CertAlias", "value": "CLEAR[<KUBERNETES_CER T_ALIAS>]" }, { "name": "CertPassword", "value": "CLEAR[<KUBERNETES_CER T_ALIAS>]" }] }] }</pre>	<pre>{ "credentials": [{ "id": "KubeCredsRef", "name": "SSLKeyStoreCredential Set", "credType": "StoreCredential", "properties": [{ "name": "StoreLocation", "value": "CLEAR[/scratch/ dritwik/view_storage/ jsons/kubernetes/ keystore.jks]" }, { "name": "StoreType", "value": "CLEAR[JKS]" }, { "name": "StorePassword", "value": "CLEAR[welcome]" }, { "name": "Certificate", "value": "FILE[<KUBERNETES_CERT _FILE_LOC>]" }] }] }</pre>

Basic Credentials	Token Credentials	Keystore Credentials
<pre> "name": "CertPassword", "value": "CLEAR[<KUBERNETES_JKS _PASSWORD>]" </pre>	<pre> "CLEAR[<KUBERNETES_JKS _PASSWORD>]" }] } </pre>	<pre> "value": "CLEAR[<KUBERNETES_CER T_ALIAS>]" }, { "name": "CertPassword", "value": "CLEAR[<KUBERNETES_JKS _PASSWORD>]" }] }] </pre>

Property description Basic Credentials

Property	UI Name	Description
Alias	Username	Username of the user going to discover Kubernetes
Password	Password	Password of the user
Certificate	Keystore Certificate	Certificate of Kubernetes API Server on Master Node. Users need to specify the text inside the certificate file if added from UI. In omcli, users need to create a Java Keystore, add certificate to that and specify the file path.
CertAlias	Certificate Alias	Alias for the Certificate. This should be unique alphanumeric string
CertPassword	Trust Store Password	Password of agent's Trust Store. This password is "welcome"

Token Credentials

Property	UI Name	Description
Token	Token	Token of the user going to discover Kubernetes
Certificate	Keystore Certificate	Refer Basic Credentials
CertAlias	Certificate Alias	Refer Basic Credentials
CertPassword	Trust Store Password	Refer Basic Credentials

Keystore Credentials

Property	UI Name	Description
StoreLocation	Store Location	Location of Client keystore. This Java Keystore file (JKS) should contain client's certificate.
StoreType	Store Type	Store type. This value is always set to "JKS"
StorePassword	Store Password	Password of the JKS file
Certificate	Keystore Certificate	Refer Basic Credentials
CertAlias	Certificate Alias	Refer Basic Credentials
CertPassword	Trust Store Password	Refer Basic Credentials

Oracle GoldenGate

Prerequisites

Oracle GoldenGate enables the continuous, real-time capture, routing, transformation, and delivery of transactional data across heterogeneous (Oracle, DB2, MySQL, SQL Server, Teradata) environments. The following prerequisites apply when discovering and monitoring Oracle GoldenGate environments.

Enable Monitoring

The first prerequisite is to enable monitoring in GoldenGate. Follow the steps below for your specific GoldenGate version and architecture.

Classic Architecture

If you are using GoldenGate Classic Architecture, you will need to add a parameter in the GLOBALS file to enable monitoring.

You must be running GoldenGate version 12.3.0.1.181120 at a minimum. This is a cumulative patch set for GoldenGate released in Jan 9, 2019.

1. Locate the GLOBALS file in the top-level GoldenGate installation directory.
2. Add the following line to this file and save the file:

```
ENABLEMONITORING UDPPORT <port> HTTPPORT <port>
```

3. Restart GoldenGate Manager.

Microservices Architecture

If you are using GoldenGate Microservices Architecture, then as part of the setup of GoldenGate using the GoldenGate Configuration Assistant, you should enable Monitoring. Once monitoring has been enabled, the Performance Metric Server will be started. This is an indication that monitoring has been enabled for GoldenGate.

OCI GoldenGate

If you are using OCI GoldenGate, no prerequisites are required. Monitoring is enabled by default.

Import Certification for GoldenGate Secure Installations

If the Oracle GoldenGate setup is secure (HTTPS), the GoldenGate certificate needs to be imported into the agent manually prior to discovery. To do this, perform the following:

1. Extract the certificate from Oracle GoldenGate.

```
openssl s_client -showcerts -connect <hostname>:<service port>
```

2. Add the Oracle GoldenGate certificate to the cloud agent's JKS.

```
<jdk>/bin/keytool -importcert -file <goldengate-certificate> -alias  
goldengate -keystore <AGENT_HOME>/agent_inst/sysman/config/montrust/  
AgentTrust.jks -storepass welcome
```

3. Bounce the cloud agent.

```
omcli stop agent ; omcli start agent
```

Oracle VM Manager

Prerequisites

The cloud agent must be deployed on the Oracle VM Manager host.

Credentials: The username and password are required to access the Oracle VM Manager console.

Example:

```
username=admin / password=admin_pw
```

Certificates:

You need to explicitly add the Oracle VM Manager Weblogic certificate to the Agent's JKS.

How to extract certificate from Oracle VM Manager:

To export the Oracle VM Manager WebLogic certificate, log in as the root user and enter the following command:

```
#!/u01/app/oracle/ovm-manager-3/ovm_upgrade/bin/  
ovmkeytool.sh          exportca >  
    <file_loc_for_certificate>
```

To import the Oracle VM Manager Weblogic certificate to the Agent Keystore, log in as an Oracle cloud agent user and enter the following command:

```
<AGENT_INSTANCE_HOME>/bin/omcli secure  
add_trust_cert_to_jks -trust_certs_loc  
    <file_loc_for_certificate> -alias <alias_name>
```

Oracle JVM Runtime

Prerequisites

Monitoring Oracle JVM Runtime can be performed in the following modes:

1. No user authentication, No SSL
2. No user authentication, SSL
3. User authentication, No SSL
4. User Authentication, SSL

SSL configuration:

You will need to import your truststore certificate into the cloud agent truststore using omcli as shown in the following example:

```
$ omcli secure add_trust_cert_to_jks -alias <Alias of cert to import>  
-trust_certs_loc <Cert file to import>
```

Microsoft Azure

Prerequisites

Before monitoring Microsoft Azure with Infrastructure Monitoring, you first need to create a *Web app/API* application registration for your Oracle Management Cloud account in Azure.

1. Log in to the Microsoft Azure portal using Global administrator credentials: .

<https://portal.azure.com/#home>

 **Note:**

You can find *Azure Active Directory*, *Subscriptions*, etc. in the *All Services* menu (or you may already have them in Favorites)

2. Under Azure services, open the "Azure Active Directory" service. Click **More services** to see the *Azure Active Directory* service.
3. Switch to the *Default Directory* if not already selected.
4. In the *Manage* section, click **App Registrations**.
5. Click **New Application Registration** and fill in details:
 - Name = Name of your application, e.g. *OMCAzureMonitoring*
 - Supported account types = Accounts in this organizational directory only (Default Directory only - Single tenant)
 - Redirect URI (optional) = Web (Leave blank--does not affect discovery.)
 - Click **Register**.
6. After clicking Register, you'll see the *Application ID* (you can later find the *Application ID* in the *Application Registration* blade). Save the *Application (client) ID* and *Directory (tenant) ID* as you'll need them later to add the Microsoft Azure entity to Oracle Management Cloud.
7. From the Manage section, click **Certificates & secrets**.
8. In the area below *Client Secrets* click **+New client secret**, and provide the Description and expiration time frame. Click **Add**.
9. The Client Secret has been created. Make note of the Client Secret value as you'll need it for discovery.

 **Note:**

You won't be able to check the value after you leave the screen. You need to create a new secret if you ever forget or lose the key you've just created.

10. Navigate Home, then to *Subscriptions*.
11. Copy the *Subscription ID* directly from the screen or select the desired Subscription (in either case, it needs to belong to the Azure Active Directory used above) and copy the *Subscription ID* from the Overview section. Save the *Subscription ID* as you'll need it for discovery.
12. Grant access to application principal to Azure resources.
There is no Oracle Management Cloud recommended practice to define the access policy and

Prerequisites

permissions for the user/application in the Azure Active Directory (tenant). Access depends on the customer security policy to allow monitoring access to the whole tenant, resource group or just individual resources. Further explanation can be found in the Azure documentation for role-based access control (RBAC).

- The easiest approach is to grant the Monitoring Reader role to the registered app at the Subscription level:
 - a. Navigate to Subscriptions.
 - b. Select the desired Subscription (it needs to belong to the Azure Active Directory used above).
 - c. Select Access control (IAM) and click Add. Fill in details for the following:
 - Role: Monitoring Reader
 - Assign access to: Azure AD user, group, or service principal (default value)
 - Select: Type in name of the application (e.g. OMCazureMonitoring as used above).
 - d. Click Save.
 - The process to grant access for Resource Groups or Resources is the same (instead of Subscription select the desired Resource Group or Resource).
 - More information on how to create the service principal can be found in Azure documentation.
- 13.** Once you've created a Web app/API application registration for your Oracle Management Cloud account in Azure, you're now ready to add the Microsoft Azure entity to Oracle Management Cloud. Use the saved values (from the Azure prerequisites above) to fill in the details in the Cloud Discovery Profile and Monitoring Credentials for Azure.
-

Apache Kafka

Prerequisites

Import Zookeeper Jar

- Place the Zookeeper jar from <Zookeeper Installation Home>/zookeeper-<version>.jar in an appropriate directory which is readable by the cloud agent user.
- Add the location where the jar is placed to the Cloud Agent classpath.lst file available under <Agent Installation Directory>/sysman/config/oracle.em.sgfm.classpath.lst
- Restart the cloud agent for the inclusion to take effect.

Import Kafka Client Jar

- Place the Kafka Client jar from <Kafka Installation Home>/libs/kafka-clients-0.10.2.1.jar in an appropriate directory which is readable by the cloud agent user
- Add the location where the jar is placed to the cloud agent Classpath.lst file available under <Agent Installation Directory>/sysman/config/oracle.em.sgfm.classpath.lst
- Restart the cloud agent for the inclusion to take effect.

JMX Configuration

Enable JMX on Kafka Brokers with no authentication

Add the following lines to <Kafka Installation Home>/bin/kafka-server-start.sh to enable JMX with no authentication.

- export JMX_PORT=<enter jmx port>
- KAFKA_JMX_OPTS="-Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false-Djava.rmi.server.hostname=<Hostname> -Dcom.sun.management.jmxremote.port=<JMX port> -Dcom.sun.management.jmxremote.rmi.port=<RMI port>"

Start the Kafka server.

 **Note:**

For a multibroker setup, provide a unique JMX port and unique broker ID for each. The listen port should be different if brokers are started on the same node. If there are duplicate broker IDs, the connection will fail with an error "Address already in use".

B

Entity Attributes and Properties

For first time discovery of your entities, the following tables provide details on the various entity properties (JSON properties/UI fields) you will need to customize. The entity properties listed are the *minimum* required for successful discovery of your entities.

Oracle JVM Runtime

Oracle JVM Runtime JSON Files and Properties

Definition File: **omc_jvm_sample_creds.json**

- **JVM host name (hostname)** - the host where the JVM application is running. It is specified during discovery in the discovery JSON or discovery UI. The value populated during discovery is the value specified by the user in discovery parameters
- **JVM application JMX port (omc_jmx_port)** - the JMX port where the JVM application is running. It is specified during discovery in the discovery JSON or discovery UI. The value populated during discovery is the value specified by the user in discovery parameters
- **JVM runtime name (omc_runtime_name)** - value determined by querying runtime MBean value Name (. The format of the value is <pid>@hostname.
- **JVM application JMX service URL (omc_jmx_service_url)** - This values is calculated from the typical remote JMX service URL format -

```
service:jmx:rmi:///jndi/rmi://<hostname>:<jmx_port>/jmxrmi
```

where the hostname and jmx port is determined from the values specified by the user in the discovery parameters.

- **JVM application class name - (omc_jvm_application_class_name)** - value determined by querying runtime MBean value SystemProperties. The value is located by the key "sun.java.command" in the list of system properties.
- **JVM application instance command line arguments (omc_jvm_app_command_line_args)** - value determined by querying runtime MBean value InputArguments. The value returned by the MBean is formatted from an array to space delimited string. This is the command line arguments specified by the user when the JVM application is started

Identifying properties

The identifying properties specified in the target model for reconciliation purposes are the following:

- host_name
- omc_jmx_port
- omc_jvm_application_class_name

Credential File: **omc_jvm_sample_creds.json**

- **user_name:** JVM username.
 - **password:** JVM password.
-

Oracle JVM Runtime UI Properties

- **Discover Using Credentials:** Discover JVM Runtime using JVM credentials (on by default).
- **Entity Name:** Name of the JVM Runtime entity appearing in the UI.
- **Host Name:** The host where the JVM application is running.
- **JMX Port Number:** The JMX port where the JVM application is running.
- **Cloud Agent:** The cloud agent monitoring the JVM application.

Monitoring Credentials

- **JMX Remote Access Username:** JVM username.
 - **JMX Remote Access Password:** JVM password.
-

Oracle VM Manager

Oracle VM Manager JSON Files and Properties

Definition File: **omc_oracle_vm_manager.json**

- **name:** Your Oracle VM Manager name. This needs to be unique across OVM Managers used
- **display name:** Name displayed in the Oracle Infrastructure Monitoring Service User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example:

America/New_York

- **omc_ovmm_console_url:** Under “value”, provide the Oracle VM Manager console URL used to connect to the installed Oracle VM Manager. The URL follows the format: `https://<ovm_host_name>:<port>/ovm` where `ovm_host_name` is a fully-qualified host name where Oracle VM Manager is installed and `port` refers to the port number on which Oracle VM Manager is listening.

Credential File: **omc_oracle_vm_manager_creds.json**

- **OVMUsername:** Under “value”, within the square brackets, provide the Oracle VM Manager console user name (default admin) to be used for monitoring.
- **OVMPassword:** Under “value”, within the square brackets, provide the Oracle VM Manager console monitoring user's password.

Do not remove the square brackets.

Oracle VM Manager UI Properties

- **Entity Name:** Name appearing in the UI.
- **Oracle VM Manager Console URL:** Oracle VM Manager console URL used to connect to the installed Oracle VM Manager. The URL follows the format: `https://<ovm_host_name>:<port>/ovm` where `ovm_host_name` is a fully-qualified host name where Oracle VM Manager is installed and `port` refers to the port number on which Oracle VM Manager is listening.
- **Cloud Agent:** The cloud agent monitoring the Oracle VM Manager application.

Monitoring Credentials

- **Admin Username:** Oracle VM Manager username.
 - **Admin Password:** Oracle VM Manager password.
-

Apache HTTP Server

Apache HTTP Server JSON Files and Properties

Definition File: **omc_generic_apache_sample.json**

- **host_name**: Host Name of the Apache HTTP Server.
 - **omc_listen_port**: Listen Port of the Apache HTTP Server.
 - **omc_httpd_conf_path**: Absolute Path of httpd.conf
 - **omc_protocol**: Protocol for connection to the Apache HTTP Server.
 - **omc_server_root**: Server Root of the Apache HTTP Server.
 - **omc_is_remote**: Indicates whether the HTTP Apache Server is local(NO) or remote(YES) - possible values: yes / no.
 - **omc_binary_home**: Absolute path of the httpd binary (Optional - default value, if not provided: \$omc_server_root/bin)
 - **omc_access_log_path**: Access Log Path (Optional)
 - **omc_error_log_path**: Error Log Path (Optional)
 - **omc_server_status_connect_host**: Server-status (Optional). This property specifies the value for the Host Name configured for /server-status connection (if different than FQDN - e.g., localhost). This needs to be specified if the connection-string (host:port) has a different host-name value than the value specified for host_name property. If specified, this value will be used to connect to Apache and retrieve data from the URI /server-status. If this property value is not specified, the default value will be the same as the host_name property value. (Optional)
-

Apache HTTP Server UI Properties

- **Entity Name**: Name of the Apache HTTP Server entity appearing in the UI. .
 - **Host Name**: Host Name of the Apache HTTP Server.
 - **Server Root**: Server Root of the Apache HTTP Server.
 - **Absolute Path of httpd.conf**: Absolute path of the Apache httpd.conf file. Note: Filename needs to be appended.
 - **Is Remote**: Is the Apache installation host different from the agent host? Yes/No
 - **Binary Home**: Absolute path of the httpd binary (Optional - default value, if not provided: <Apache Home>/bin or /usr/bin)
 - **Protocol**: Protocol for connection to the Apache HTTP Server.
 - **Listen Port**: Listen Port of the Apache HTTP Server.
 - **Server Status Connection Hostname**: Server-status (Optional). This property specifies the value for the Host Name configured for /server-status connection (if different than FQDN - e.g., localhost). This needs to be specified if the connection-string (host:port) has a different host-name value than the value specified for host_name property. If specified, this value will be used to connect to Apache and retrieve data from the URI /server-status. If this property value is not specified, the default value will be the same as the host_name property value. (Optional)
 - **Cloud Agent**: Cloud agent monitoring the Apache HTTP Server.
-

MySQL Database

MySQL Database JSON Properties and Files

Definition File: **omc_mysql_db_sample.json**

- **name:** Your MySQL database name.
- **display name:** Name displayed in the Oracle Infrastructure Monitoring Service User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: `America/New_York`
- **url:** Under "value", provide the connection URL used to connect to the installed MySQL database. The URL follows the format:
`jdbc:mysql://<host_name>:<port>/mysql` where `host_name` is a fully-qualified host name where MySQL Database is installed and `port` is the MySQL Database port defined at installation time.
- **hostname:** Under "value", provide the fully-qualified host name where MySQL Database is installed.
- **is_cluster:** (TRUE/FALSE) Specifies whether or not you are adding a MySQL Cluster Database.

Definition File: **omc_mysql_db_cluster_sample.json**

- **url**
`jdbc:mysql://host1:<port1>,host2:<port2>/dbname`
 where:
 - Host 1 and Host 2 would be same in case of Single Host Cluster
 - Instance 1 / Node 1
`instance_name: <host1>.mycompany.com:<port1>`
 - Instance 2 / Node 2
`instance_name: <host2>.mycompany.com:<port2>`
- **jdbcdriver:** `com.mysql.jdbc.Driver`
- **MachineName:** Your MySQL Database Host Name
- **Is Cluster:** `true/false`
- **capability:** `monitoring`

Credential File: **omc_mysql_creds.json**

- **DBUserName:** Under "value", within the square brackets, provide the MySQL database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
- **DBPassword:** Under "value", within the square brackets, provide the MySQL database monitoring user's password.

Do not remove the square brackets.

MySQL Database UI Properties

- **Entity Name:** Name displayed in the Oracle Management Cloud console.
- **JDBC URL:** The connection URL for the MySQL Database. The URL follows the format:
`jdbc:mysql://<host_name>:<port>/mysql` where `host_name` is a fully-qualified host name where MySQL Database is installed and `port` is the MySQL Database port defined at installation time.
- **Host Name:** The fully-qualified host name where MySQL Database is installed.
- **Cloud Agent:** Agent monitoring the host on which the database is installed.

Monitoring Credentials

- **Username:** MySQL Database user name to be used for monitoring.
 - **Password:** MySQL Database user password.
-

Oracle Database System (single instance)

Oracle Database System (single instance) JSON Properties and Files

Definition File: **omc_oracle_db_system_SI.json**

- **name**: Your Oracle Database Entity Name. Will also be used for the Database System name
- **displayName**: Your Oracle Database Entity Display Name. Will also be used for the Database System display name.
- **timezoneRegion**: Your timezone
- **omc_dbsys_config**: Configuration for DB System. Here, it is SI.
- **omc_dbsys_name_qualifier**: Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **host_name**: Name of the listener host that will be used to create the connect string to the database (host:port:SID or host:port:ServiceName)
- **omc_dbsys_port**: Listener port number used for connection requests
- **omc_dbsys_connect_type**: Specify type of connection: SID or Service Name
- **omc_dbsys_connect_value**: The value of the SID or Service Name
- **omc_dbsys_lsnr_alias**: Value of Listener Alias
- **omc_dbsys_home**: Oracle Home directory of the Listener
- **capability**: monitoring

Credential File: **omc_oracle_db_system_creds_SI_local.json**

- **DBUserName** : Your Database User Name
- **DBPassword** : Your Database Password
- **DBRole** : Your Database User Role. Default : Normal
- If Remote:

Credential File: **omc_oracle_db_system_creds_SI_with_SSH.json**

- **SSHUserName**: Your SSH user used to remotely logon to the listener host
 - **SSHUserPassword** : Your SSH host Password
 - **SSH_PVT_KEY**: Path of your private key file. This private key is optional if the keys are generated at default location <user home>/ .ssh
 - **sshdPort**: SSH port
-

Oracle Database System (single instance) UI Property Fields

- **Entity Name:** Your Oracle Database entity name. This name will also be used for the database system name
- **Configuration:** Configuration for database system: Single Instance or RAC
- **Name Prefix:** Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **Host Name:** Name of the listener host that will be used to create the connect string to the database (host:port:SID or host:port:ServiceName)
- **Port:** Listener port number used for connection requests
- **Connect Type:** Type of connection: SID or Service Name
- **Connect Value:** The value of the SID or Service Name
- **Listener Alias:** Value of Listener Alias
- **Listener Oracle Home:** Oracle Home directory of the Listener
The *Listener Oracle Home* field in the UI is the Oracle Home of the listener configured for that database. The Oracle Home for the listener may or may not be the same Oracle Home as the database as illustrated by the following example.
The following example shows two discrete database instances (prod_1 and test_1) in two separate Oracle Homes:
Oracle Home 1: /u01/app/oracle/product/19.0.0/prod_1
Oracle Home 2: /u01/app/oracle/product/19.0.0/test_1
Because both instances are configured with the listener in Oracle Home 1, to discover the test_1 instance (in Oracle Home 2) you would enter /u01/app/oracle/product/19.0.0/prod_1 in the *Listener Oracle Home* field.
- **Cloud Agent:** Cloud agent monitoring the database system.

Monitoring Credentials

- **Username:** Your Database User Name.
- **Password:** Your Database Password.
- **Database Role:** Your Database User Role (NORMAL/SYSDBA). Default is Normal.

SI with ASM (ASM Credentials)

- **Username:** Database user (ASM user name) that will be used by the cloud agent to connect to ASM.
- **Password:** Your ASM Password
- **Role:** Your ASM User role

Cloud Agent is not on the Cluster Host (Host SSH Credentials)

- **SSH Username:** Your SSH user used to remotely log on to the listener host.
 - **SSH Password:** Your SSH host Password.
 - **SSH Private Key:** Path of your private key file.
 - **SSH Port:** Your SSH port.
-

Oracle Database System (RAC)

Oracle Database System (RAC) JSON Properties and Files

Definition File: **omc_oracle_db_system_RAC.json**

- **name**: Your Oracle Database Entity Name. Will also be used for the Database System name
- **displayName**: Your Oracle Database Entity Display Name. Will also be used for the Database System display name.
- **timezoneRegion**: Your timezone.
- **omc_dbsys_config**: Configuration for DB System. Here, it is RAC.
- **omc_dbsys_name_qualifier**: Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **omc_db_system_scan_name**: Name of the SCAN Listener or SCAN VIP
- **omc_dbsys_port**: Port number on which the SCAN listener is listening for connections
- **omc_dbsys_connect_type**: Service Name
- **omc_dbsys_connect_value**: Service Name registered with the listener which is used to connect to the database
- **omc_dbsys_home**: Oracle home directory for the Oracle Grid Infrastructure
- **capability**: monitoring

Credential Files

omc_oracle_db_system_creds_RAC_local_with_ASM.json

omc_oracle_db_system_creds_RAC_with_SSH_with_ASM.json

omc_oracle_db_system_creds_RAC_local_without_ASM.json

omc_oracle_db_system_creds_RAC_with_SSH_without_ASM.json

Credential properties:

- **DBUserName** : Your Database User Name
- **DBPassword** : Your Database Password
- **DBRole** : Your Database User Role. Default : Normal

If ASM is also to be discovered:

- **user_name**: Your ASM User Name
- **password**: Your ASM Password
- **role**: Your ASM User role

If Remote:

- **SSHUserName**: Your SSH user used to remotely log onto the listener host
 - **SSHUserPassword** : Your SSH host Password. Optional , if there is a passwordless SSH setup. In this case, provide a private key field
 - **SSH_PVT_KEY**: Path of your private key file. This private key is optional if the keys are generated at default location <user home>/ .ssh
 - **sshdHost**: Your Cluster Host Name
 - **sshdPort**: SSH port
-

Oracle Database System (RAC) UI Fields

- **Entity Name:** Your Oracle Database Entity Name. Will also be used for the Database System name.
- **Name Prefix:** Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **SCAN Name:** Name of the SCAN Listener or SCAN VIP.
- **SCAN Port:** Port number on which the SCAN listener is listening for connections.
- **Service Name:** Service Name registered with the listener which is used to connect to the database.
- **Grid Home:** Oracle home directory for the Oracle Grid Infrastructure.
- **Cloud Agent:** Cloud agent used to monitor the cluster.

Monitoring Credentials

- **Username:** Your Database User Name.
- **Password:** Your Database Password.
- **Database Role:** Your Database User Role (NORMAL/SYSDBA). Default is Normal.

RAC with ASM (ASM Credentials)

- **Username:** Database user (ASM user name) that will be used by the cloud agent to connect to ASM.
- **Password:** Your ASM Password
- **Role:** Your ASM User role

Cloud Agent is not on the Cluster Host (Host SSH Credentials)

- **SSH Username:** Your SSH user used to remotely log on to the listener host.
 - **SSH Password:** Your SSH host Password.
 - **SSH Private Key:** Path of your private key file.
 - **SSH Public Key:** Path of your public key file.
 - **SSH Host Name:** Your Cluster Host Name.
 - **SSH Port:** Your SSH port.
-

Oracle Database (Single Instance)

Oracle Database (Single Instance) JSON Properties and Files

Definition File: **omc_oracle_db_sample.json**

- **name:** Your Oracle database name.
- **display name:** Name displayed in the Oracle Infrastructure Monitoring Service User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host name:** Under “value”, provide the fully-qualified host name where the Oracle Database is installed.
- **port:** The port that the listener is using. Check the listener.ora file or output of lsnrctl status <listener name>. For example, “1521”.
Under “value”, list the Oracle database port.
- **sid:** The instance name of the database. You can determine the SID using the following command:
ps -ef | grep pmon or show parameter instance_name if using SQL*Plus. For example, “instance_name” could be orcl.
Under “value”, list the Oracle database SID.

Credential File: **omc_oracle_db_creds.json**

IMPORTANT! Only change the properties in the bulleted list below. DO NOT change the values for id, name, or credType. Values for these parameters should be SQLCreds, SQLCreds, and DBCreds respectively.

- **DBUserName:** The name of a database user who has the necessary privileges on the underlying V\$views such as moncs, or the monitoring user.
Under “value”, within the square brackets, provide the Oracle database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
- **DBPassword:** The password of your monitoring user.
Under “value”, within the square brackets, provide the Oracle database monitoring user's password. Do not remove the square brackets.
- **DBRole:** The role that the MONCS user will use when logging into the database (eg. Normal, Sysdba, Sysasm etc. Under normal circumstances, Normal should be enough. However, SYSDBA might be required for a Dataguard database.



Note:

CDB/PDB also applies to single instance databases.

Oracle Pluggable Database (PDB) (Standalone)

Oracle Pluggable Database (PDB) (Standalone) JSON Properties and Files

Definition File: **omc_oracle_pdb_sample.json**

- **displayName:** This is Oracle Pluggable Database (PDB) Entity Display Name which is displayed on Infrastructure Monitoring UI
- **timezoneRegion:** Time Zone Example: PDT, GMT etc
- **host_name :** Fully-qualified Host Name for the Oracle Pluggable Database (PDB)
- **omc_pdb_tbsp_port:** Oracle Pluggable Database (PDB) port
- **omc_pdb_tbsp_service_name:** Oracle Pluggable Database (PDB) Service Name

Credential File: **omc_oracle_pdb_cred_sample.json**

- **DBUserName:** Oracle Pluggable Database (PDB) username
 - **DBPassword:** Oracle Pluggable Database (PDB) user's password
-

Oracle Database (Real Application Clusters)

Oracle Database (Real Application Clusters) JSON Properties and Files

Definition File: **omc_oracle_dbRAC_sample.json**

- **name:** Your Oracle database name.
- **display name:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host name:** Under “value”, provide the fully-qualified host name where the Oracle Database is installed.
- **port:** Under “value”, list the Oracle database port.
- **service_name:** Under “value”, list the Oracle database service name.

 **Note:**

CDB/PDB also applies to RAC databases.

Credential File: **omc_oracle_dbRAC_sample_creds.json**

- **DBUserName:** The name of a database user who has the necessary privileges on the underlying V\$views such as db\$snmp, or the monitoring user. Under “value”, within the square brackets, provide the Oracle database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
 - **DBPassword:** The password of your monitoring user. Under “value”, within the square brackets, provide the Oracle database monitoring user's password. Do not remove the square brackets.
-

Oracle Automation Storage Management

Oracle Automation Storage Management JSON Files and Properties

Definition File: **omc_oracle_asm_sample.json**

- **name:** Your Oracle ASM entity name.
- **displayName:** Your Oracle ASM entity display name which is displayed on the Oracle Infrastructure Monitoring user interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **machine_name:** Under “value”, provide the fully-qualified machine name where the Oracle ASM is installed.
- **host_name:** Under “value”, provide the fully-qualified host name where the Oracle ASM is installed.
- **port:** Under “value”, list the Oracle ASM port.
- **sid:** Under “value”, list the Oracle ASM SID.

Credential File: **omc_oracle_asm_sample_creds.json**

- **user_name:** Under “value”, within the square brackets, provide the Oracle ASM user name to be used for monitoring.
- **password:** Under “value”, within the square brackets, provide the Oracle ASM monitoring user's password.

Do not remove the square brackets.

Oracle Database Listener

Oracle Database Listener JSON Files and Properties

Definition File: **omc_oracle_db_listener_sample.json**

- **displayName:** This is Oracle Database Listener Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Fully-qualified Host Name where the Oracle Database Listener is installed.
- **port:** Oracle Database Listener port.
- **trace_dir_path:** Trace Log Files directory absolute path; optional parameter, define it if you are also using Oracle Log Analytics.
- **log_dir_path:** Alert Log Files directory absolute path; optional parameter, define it if you are also using Oracle Log Analytics.
- **lsnr_alias:** Oracle Database Listener Alias.

Credential Files:

omc_oracle_db_listener_local_credless.json

omc_oracle_db_listener_remote_ssh_sample.json

omc_oracle_db_listener_remote_ssh_sample_creds.json

omc_oracle_db_listener_creds.json

- **displayName:** This is Oracle Database Listener Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **use_ssh:** Use Secure Shell protocol. (true/false).
True :- Set to true when performing remote monitoring.
False:- Set to false when performing local monitoring (agents 1.30 and lower). For agents 1.31 and greater, the use_ssh property is optional for local monitoring.
- **lsnr_port:** Your Oracle Database Listener Port.
- **lsnr_protocol:** Your Listener Protocol
- **oracle_home:** Your Oracle Listener ORACLE_HOME.
- **log_dir_path:** Alert Log Files directory absolute path; optional parameter, define it if you are also using Oracle Log Analytics.
- **trace_dir_path:** Trace Files directory absolute path.
- **lsnr_alias:** Your Oracle Database Listener Alias.
- **SSHUserName:** SSH host user name, on the host where the listener is installed.
- **SSHUserPassword:** SSH host user password
- **SSH_PVT_KEY:** Path of your private key file - Not required if a password is provided or SSH keys are available in the default location.
- **sshdPort:** SSH port.

 **Note:**

You must use a host user with SSH configured and enabled. Only password-based SSH is supported.

Oracle Database Listener Cluster

Oracle Database Listener Cluster JSON Files and Properties

Definition File: **omc_oracle_db_listener_cluster_sample.json**

- **displayName**: This is Oracle Database Listener Cluster Entity Display Name which is displayed on Infrastructure Monitoring UltimatezoneRegion: Time Zone Example: PDT, GMT
- **host_name**: : Fully-qualified Host Name where the Oracle Database Listener Cluster is installed.
- **lsnr_alias** : Oracle Database Listener Cluster Alias
- **crs_home** : Absolute Path of the CRS HOME / GRID HOME

Credential Files

omc_oracle_db_listener_cluster_credless_sample.json

omc_oracle_db_listener_cluster_sample_cred.json

- **sshdPort**: SSHD Port on Remote host to Listen to Remote Cluster Listener
 - **SSHUserName**: SSH Host User Name
 - **SSHUserPassword**: SSH Host User Password
 - **SSH_PVT_KEY** : Location of the SSH private key copied from the remote machine where Cluster Listener is installed.
 - **SSH_PUB_KEY** : Location of the SSH public key copied from the remote machine where Cluster Listener is installed.
 - **sshdHost** : Host Name where Oracle Database Listener Cluster is installed.
-

Microsoft SQL Server Database

Microsoft SQL Server Database JSON Files and Properties

Definition File: **omc_sqlserver_db_sample.json**

- **name:** Your Microsoft SQL Server database name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **url:** Under “value”, provide the connection URL for the MS SQL Server database. The URL follows the formats:
 - Connect to default instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>
```

- Connect to named instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>\
<Instance Name> (or)
jdbc:sqlserver://<Fully-qualified SQL Server Database Host
Name>;instanceName=<instance-name>
```

- Connect to instance by specifying port.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host
Name>:<SQL Server Database Port>
```

- Connecting with SSL encryption.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>\
<Instance
Name>;encrypt=true;trustServerCertificate=false;trustStore=<Path to
trust store file>;trustStorePassword=<trust store password>
```

See the prerequisites section for details on setting up the certificates and trust store.

Credential File: **omc_sqlserver_creds.json**

- **DBUserName:** Under “value”, within the square brackets, provide the MS SQL Server database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
- **DBPassword:** Under “value”, within the square brackets, provide the MS SQL Server database monitoring user's password.

Do not remove the square brackets.

Microsoft SQL Server Database UI Fields

- **Entity Name:** Name displayed in the Oracle Management Cloud console.
- **JDBC URL:** The connection URL for the MS SQL Server database. The URL follows the formats:

- Connect to default instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>
```

- Connect to named instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>\<Instance Name> (or)
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>;instanceName=<instance-name>
```

- Connect to instance by specifying port.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>:<SQL Server Database Port>
```

- Connecting with SSL encryption.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>\<Instance Name>;encrypt=true;trustServerCertificate=false;trustStore=<Path to trust store file>;trustStorePassword=<trust store password>
```

See the prerequisites section for details on setting up the certificates and trust store.

- **Cloud Agent:** Agent monitoring the host on which the database is installed.

Monitoring Credentials

- **Username:** MS SQL Server database user name to be used for monitoring.
 - **Password:** MS SQL Server database monitoring user's password.
-

Oracle NoSQL

Oracle NoSQL JSON Files and Properties

Definition File: **omc_nosql_db_sample.json**

- **name:** Your Oracle NoSQL Database entity name.
 - **displayName:** This is Oracle NoSQL Database Entity Display Name which is displayed on Infrastructure Monitoring UI.
 - **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
 - **host_name:** Under "value", provide the fully-qualified host name where the Oracle NoSQL Database is installed.
 - **omc_port:** Under "value", list the NoSQL Database port.
 - **omc_url:** Connection URL to connect to the installed Oracle NoSQL database. Comma separated host:port of all Nosql nodes of a store. example:
<host_name>:<port>,<host_name>:<port>
 - **omc_nosql_java_home:** Under "value", list the Java home of NoSQL database or any Java home of version above 1.8 .
 - **omc_kv_home:** Under "value", list the KV home of the NoSQL database.
-

Oracle NoSQL UI Fields

- **Entity Name:** Your Oracle NoSQL Database entity name.
 - **Host Name:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
 - **URL:** The NoSQL database connection URL.
 - **Port:** The NoSQL database port.
 - **Store Name:** Oracle NoSQL Database store name.
 - **NoSQL Java Home:** Java home of NoSQL database or any Java home of version above 1.8 .
 - **KV Home Location:** KV home of the NoSQL database.
 - **Cloud Agent:** Agent monitoring the host on which the database is installed.
-

MongoDB Database

MongoDB Database JSON Files and Properties

Description File: **omc_mongodb_sample.json**

- **name:** Your MongoDB database name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under "value", provide the fully-qualified host name where the MongoDB Database is installed.
- **port:** Under "value", list the MongoDB database port.
- **database_name:** Under "value", list the MongoDB database name.

Credential File: **omc_mongodb_creds.json**

- **DBUserName:** Under "value", within the square brackets, provide the MongoDB database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
- **DBPassword:** Under "value", within the square brackets, provide the MongoDB database monitoring user's password.

Do not remove the square brackets.

MongoDB Database UI Fields

- **Entity Name:** Your MongoDB database name.
- **Host Name:** the fully-qualified host name where the MongoDB Database is installed.
- **Port:** MongoDB database port.
- **Database Name:** MongoDB database name.
- **Cloud Agent:** Cloud agent monitoring the host on which MongoDB is installed.

Monitoring Credentials

- **Username:** MongoDB database user name to be used for monitoring.
 - **Password:** MongoDB database monitoring user's password.
-

Tomcat

Tomcat JSON Files and Properties

Definition File: **omc_tomcat_secure_sample.json**

- **name:** Your Tomcat name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under "value", provide the fully-qualified Tomcat host name where the Tomcat entity is installed.
- **jmx_port:** Under "value", provide the JMX port used by the Tomcat entity.

Credential Files

omc_tomcat_secure_creds.json

- **jmx_username:** Under "value", within the square brackets, provide the Tomcat user name. Leave this field blank and still include the credential JSON file for credential-less discovery. Do not remove the square brackets.
- **jmx_password:** Under "value", within the square brackets, provide the Tomcat user name password. Leave this field blank and still include the credential JSON file for credential-less discovery.

omc_tomcat_secureSSL_creds.json

- **ssl_trust_store:** Under "value", within the square brackets, provide the full path to the Cloud Agent truststore, AgentTrust.jks. For example, <agent base directory>/sysman/config/montrust/AgentTrust.jks
- **ssl_trust_store_password:** Under "value", within the square brackets, provide the Cloud Agent truststore password, the default is "welcome".

Do not remove the square brackets.

 **Note:**

To add a Tomcat entity that does not require credentials, simply add the entity without any credentials. And, if you do not provide any credentials, make sure input JSON file also does not contain any references to credentials.

To add a Tomcat entity without credentials, you will still need to provide the credentials file (omc_tomcat_secure_creds.json) but keep the `jmx_username` value blank, as shown in the following example.

```
{
  "credentials": [{
    "id": "TomcatCredsNormal",
    "name": "tomcat_creds",
    "credType": "TomcatCreds ",
    "properties": [{
      "name": "jmx_username",
      "value": "CLEAR[]"
    }, {
      "name": "jmx_password",
      "value": "CLEAR[]"
    }
  ]
}]
}
```

Tomcat JSON Files and Properties

 **Note:**

Beginning with Oracle Management Cloud 1.30, Tomcat discovery will always use the Agent Trust Store. User-provided SSL Trust Store will no longer be accepted.

Tomcat UI Fields

- **Discover Using Credentials:** Discover Tomcat using Tomcat credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** The fully-qualified host name where the Tomcat entity is installed.
- **JMX Port Number:** The JMX port used by the Tomcat entity.
- **Cloud Agent:** The cloud agent monitoring the host where Tomcat is installed.

Monitoring Credentials

- **JMX Username:** The Tomcat user name.
- **JMX Password:** The Tomcat user name password.

WebLogic Domain /WebLogic Server

WebLogic Domain /WebLogic Server JSON Files and Properties

Definition File: **omc_weblogic_domain_sample.json**

 **Note:**

When you add a WebLogic Domain entity (requiring credentials), because Oracle Management Cloud connects to the WebLogic Admin Server, all WebLogic Clusters and WebLogic Servers that are part of that domain are automatically discovered. There's no need to add them separately.

- **name:** Your WebLogic Domain name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **port:** Under "value", provide the port used for WebLogic Admin Server(Console).
- **protocol:** Under "value", provide the protocol used for WebLogic Server - For example: t3
- **admin_server_host:** Under "value", provide the fully-qualified host name where the WebLogic Admin Server is installed.

Credential File: **omc_weblogic_domain_sample_creds.json**

- **username:** Under "value", within the square brackets, provide the WebLogic Domain monitoring user name. You must have defined this user in the Prerequisite Tasks step.
- **password:** Under "value", within the square brackets, provide the WebLogic Domain monitoring user's password.

Do not remove the square brackets.

WebLogic Domain /WebLogic Server UI Fields

- **Entity Name:** Your WebLogic Domain name.
- **Port:** The port used for WebLogic Admin Server(Console).
- **Protocol:** The protocol used for WebLogic Server - For example: t3
- **Administration Server Host:** the fully-qualified host name where the WebLogic Admin Server is installed.
- **Discover Coherence:** (True/False) Specify whether Coherence Clusters deployed on the Weblogic domain should be discovered. This option is set to True by default. Turn this option off when discovering a SOA Suite domain.
- **Cloud Agent:** Cloud agent monitoring the host where WebLogic is installed.

Monitoring Credentials (WebLogic Server Credentials)

- **Username:** WebLogic Server user name.
 - **Password:** WebLogic Server username password.
-

Docker Engine/Docker Container

Docker Engine/Docker Container JSON Files and Properties

Definition Files

omc_docker_engine_sample.json (used without the omc_docker_engine_sample_creds.json)

omc_docker_engine_secure_sample.json (used with the omc_docker_engine_sample_creds.json)

- **name:** Your Docker Engine name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **BaseURI:** Under "value", provide the base URL for REST API invocations.
- **host_name:** Under "value", provide the fully-qualified host name where the Docker Engine is installed.

Credential File: **omc_docker_engine_sample_creds.json**

- **StoreLocation:** Under "value", within the square brackets, provide the full path to the location of the keystore file. You must have configured this entity security in the Prerequisite Tasks step. For example, <agent home>/sysman/config/montrust/AgentTrust.jks
Note that in this release only jks file types are supported.
- **StorePassword:** Under "value", within the square brackets, provide the keystore password to access the jks file.
Note that in this release only jksfile types are supported.

Do not remove the square brackets.

Docker Engine/Docker Worker UI Fields

- **Discover Using Credentials:** Discover using Docker Engine credentials (on by default).
- **Entity Name:** Your Docker Engine/Container name.
- **Base URL:** The base URL for REST API invocations.
- **Host Name:** The fully-qualified host name where the Docker Engine/Container is installed.
- **Swarm ID:** Unique identifier of the Docker Swarm containing the Docker Engine/Container.
- **Cloud Agent:** Cloud agent monitoring the host where the Docker Engine/Container is running.

Monitoring Credentials (Docker Engine Credentials)

- **Store Location:** The full path to the location of the keystore file.
 - **Store Password:** The keystore password to access the jks file.
 - **Store Type:** Currently, only JKS is supported.
-

Docker Swarm

Docker Swarm and Worker JSON Files and Properties

Entity JSONs for Docker Swarm:

Adding Non Secure Docker Swarm Target

Definition File: **Add_Entity_Docker_Swarm_Non_Secure.json**

Adding 1WAY Docker Swarm Target

Definition File: **Add_Entity_Docker_Swarm_1way_SSL.json**

Adding 2WAY Docker Swarm Target

Definition File: **Add_Entity_Docker_Swarm_2way_SSL.json**

Credential File: **Docker_Swarm_Secure_Credentials.json**

Entity JSONs for Docker Worker Engines:

Adding Non Secure Docker Worker Engine

Definition File: **Add_Entity_Worker_Docker_Engine_Non_Secure.json**

Adding 1WAY Docker Worker Engine

Definition File: **Add_Entity_Worker_Docker_Engine_1way_SSL.json**

Adding 2WAY Docker Worker Engine

Definition: **Add_Entity_Worker_Docker_Engine_2way_SSL.json**

Credential File: **omc_docker_engine_sample_creds.json**

For properties that should be updated, replace any text inside brackets <> excluding these brackets with your values according the legend inside <>

Examples of Base URLs:

NON SECURE MODE - `http://<hostname>:<port>/` (Rest API URL for Invocation)

SECURE MODE - `https://<hostname>:<port>/` (Rest API URL for Invocation)

For Basic Authentication:

"credentialRefs":["DockerSwarmCredRef"]

 **Note:**

The same Docker Engine credential JSON is used for Worker Engines.

For secure mode, in addition to configuring the JSONS, you need to add the Docker truststore certificate(CA certificate) to the Cloud Agent default truststore (\$EMSTATE/sysman/config/montrust/AgentTrust.jks). To do so, run the following command:

```
omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc
<certificate location> -alias dockercertificate
```

Example:

In the following example, slce03.cer is the CA certificate.

```
omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc /
home/sandepai/slce03.cer -alias dockercertificate
```

To fetch the Docker Swarm ID, do a GET on LEADER_BASE_URL/swarm

For example, do a GET on `http://myserver.mycompany.com:4243/swarm`

Docker Swarm UI Fields

- **Discover Using Credentials:** Discover using Docker Swarm credentials (on by default).
- **Entity Name:** Your Docker Swarm name.
- **Base URL of Swarm Leader:** The base URL of the Swarm Leader for REST API invocations.
- **Host Name:** The fully-qualified host name where the Docker Swarm is installed.
- **Cloud Agent:** Cloud agent monitoring the host where the Docker Swarm is running.

Monitoring Credentials (Docker Swarm Credentials)

- **Store Location:** The full path to the location of the keystore file.
 - **Store Password:** The keystore password to access the JKS file.
 - **Store Type:** Currently, only JKS is supported.
-

Xen Virtual Platform/Xen Virtual Server

Xen Virtual Platform/Xen Virtual Server JSON Files and Properties

The Xen Virtual Server is automatically discovered when you discover a Xen Virtual Platform.

Definition File: **omc_xen_virtual_platform_sample.json**

- **name:** Your Xen Server name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under “value”, provide the fully-qualified host name where the Xen Server is installed.

Credential Files

omc_xen_virtual_platform_with_root_creds.json

Use this credentials JSON file if using root as a monitoring user.

- **password:** Under “value”, within the square brackets, provide the root password.

omc_xen_virtual_platform_with_sudo_creds.json

Use this credentials JSON file if using a user with SUDO privileges as a monitoring user.

- **username:** Under “value”, within the square brackets, provide the user name with SUDO privileges to be used for monitoring. You must have defined this user in a previous step.
- **password:** Under “value”, within the square brackets, provide the monitoring SUDO user password.

omc_xen_virtual_platform_with_ssh_keys.json

Use this credentials JSON file if using SSH keys credentials.

- **SshPublicKey:** Under “value”, within the square brackets, provide the Public Key.
- **SshPrivateKey:** Under “value”, within the square brackets, provide the Private Key.

omc_xen_virtual_platform_with_sudo_ssh_keys.json

Use this credentials JSON file if using a user with SUDO privileges as a monitoring user as well as SSH keys.

- **username:** Under “value”, within the square brackets, provide the user name with SUDO privileges to be used for monitoring. You must have defined this user in a previous step.
- **password:** Under “value”, within the square brackets, provide the monitoring SUDO user password.
- **SshPublicKey:** Under “value”, within the square brackets, provide the Public Key.
- **SshPrivateKey:** Under “value”, within the square brackets, provide the Private Key.

Do not remove the square brackets.

Leave the rest of the fields unchanged.

Xen Virtual Platform/Xen Virtual Server UI Fields

- **Entity Name:** Your Xen Server name.
- **Host Name:** The fully-qualified host name where the Xen Server is installed.
- **Cloud Agent:** Cloud agent monitoring the host where the Xen Server is running.

Monitoring Credentials*Host Credentials*

- **Username:** Your XEN username.
- **Password:** Your XEN User Password
- **Privilege Type:** Your privilege type.
- **Privileged User:** Is this a privileged user. True/False

Host Credentials with Sudo Privileges

- **Username:** The user name with SUDO privileges to be used for monitoring.
- **Password:** The monitoring SUDO user password.
- **Privilege Type:** Your host login privilege type. Sudo is set by default.
- **Privileged User:** Is this a privileged user. True/False
- **Privilege Command:**
- **Run As:** Your RunAs Username
- **Is requiretty enabled for the user:** Enable pseudo terminal.

SSH Key Credentials

- **Username:** Your Xen username.
- **SSH Public Key:** Your public SSH key.
- **SSH Private Key:** Your private SSH key.
- **Passphrase:** Your SSH key passphrase.
- **Privilege Type:** Your privilege type.
- **Privileged User:** Is this a privileged user. True/False

SSH Key Credentials with Sudo Privileges

- **Username:** Your SUDO username.
 - **Password:** Your SUDO user password
 - **SSH Public Key:** Your public SSH key.
 - **SSH Private Key:** Your private SSH key.
 - **Passphrase:** Your SSH key passphrase
 - **Privilege Type** Your privilege type.
 - **Privileged User:** Is this a privileged user. True/False
 - **Privilege Command**
 - **Run As:** Your RunAs Username
 - **Is requiretty enabled for the user:** Enable pseudo terminal.
-

Traffic Director Instance (OTD)

Traffic Director Instance (OTD) JSON Files and Properties

Definition File: **omc_oracle_otd_cluster_sample.json**

 **Note:**

This applies specifically for OTD 11. For OTD 12, OTD will be automatically discovered as part of the WebLogic Domain discovery

- **name:** Your OTD 11g name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **admin_host:** Under "value", provide the fully-qualified Host Name where the OTD 11g Administration Server is installed.
- **admin_port:** Under "value", provide the OTD 11g Administration Server listening port.
- **admin_oracle_home:** Under "value", provide the absolute path of the OTD 11g Administration Server Oracle Home location.
- **config_name:** Under "value", provide the OTD 11g configuration name.

Credential File: **omc_oracle_otd_cluster_sample_creds.json**

 **Note:**

This applies specifically for OTD 11. For OTD 12, OTD will be automatically discovered as part of the WebLogic Domain discovery.

- **admin_user_name:** Under "value", within the square brackets, provide the OTD 11g Administration Server user name. You must have identified this user in the Prerequisite Tasks step.
- **admin_password:** Under "value", within the square brackets, provide the OTD 11g Administration Server password.
- **snmp_comm_string:** Under "value", within the square brackets, provide the community string used in SNMP subagent. The default value is "public".

Do not remove the square brackets.

Traffic Director Instance (OTD) UI Fields

12c or later

- **Entity Name:** The OTD entity name that is displayed on Infrastructure Monitoring UI.
- **Port:** OTD Administration Server listening port
- **Protocol:** t3 or t3s
- **Administration Server Host:** Fully-qualified host name where the OTD Administration Server is installed.
- **Discover Coherence:** Coherence Clusters deployed on the domain will be discovered. Turn this option off when discovering a SOA Suite Domain.
- **Cloud Agent:** Cloud agent monitoring the host where the OTD Administration Server is installed.

Monitoring Credentials (WebLogic Server Credentials)

- **Username:** OTD Administration Server username.
- **Password:** OTD Administration Server password.

11g

- **Entity Name:** OTD 11g Entity Display Name which is displayed on Infrastructure Monitoring UI
- **Administration Server Host Name:** Fully-qualified Host Name where the OTD 11g Administration Server is installed.
- **Administration Server Listen Port:** OTD 11g Administration Server listening port.
- **Configuration Name:** OTD 11g configuration name
- **Administration Server Oracle Home:** Absolute path of the OTD 11g Administration Server oracle home location
- **Cloud Agent:** Cloud agent monitoring the host where the OTD 11g Administration Server is running.

Monitoring Credentials (OTD 11g Administration Server Credentials)

- **Administration Username:** OTD 11g Administration Server username
 - **Administration Password:** OTD 11g Administration Server password
 - **SNMP Community String:** Community string used in SNMP agent. Default value is "public".
-

Oracle HTTP Server (OHS)

Oracle HTTP Server (OHS) Files and Properties

Definition File: **omc_oracle_apache_sample.json**

- **host_name**: Host Name of the Oracle HTTP Server
- **port**: Port of the Oracle HTTP Server
- **ohs_home**: Absolute path of the Instance Home (11g)/ Absolute Path of the Domain Home (12c)
- **component_name**: Component Name
- **protocol**: Protocol for connecting to the Oracle HTTP Server
- **config_path**: httpd.conf file directory path - file name not to be appended.
- **oracle_home**: Absolute path of the Oracle Home
- **version**: Version of OHS installed.

Credential Files

omc_oracle_apache_sample_creds_ohs12.json

Use this credential JSON file if you are running OHS 12

- **nm_user_name**: Node Manager username
- **nm_password**: Node Manager password

omc_oracle_apache_sample_creds_ohs11.json

Use this credential JSON file if you are running OHS 11 (optional)

- **HostUserName**: Host username
 - **HostPassword**: Host password
-

Oracle HTTP Server (OHS) UI Fields

- **Entity Name**: Name of your Oracle HTTP Server.
 - **Host Name**: Host Name of the Oracle HTTP Server
 - **Oracle Home**: Absolute path of the Oracle Home.
 - **Instance Home(11g) / Domain Home**: Absolute path of the Instance Home (11g)/Absolute path of the Domain Home (12c and later)
 - **Component Name**: Oracle HTTP Server component name.
 - **Version**: Oracle HTTP Server installed version.
 - **Configuration Path**: httpd.conf file directory path.
 - **Listen Port**: Port of the Oracle HTTP Server.
 - **Protocol**: Protocol used to connect to the Oracle HTTP Server. (HTTP/HTTPS)
 - **Cloud Agent**: Cloud agent monitoring the host where Oracle HTTP Server is installed.
-

Cisco Catalyst Ethernet Switch

Cisco Catalyst Ethernet Switch JSON Files and Properties

Definition File: **omc_cisco_eth_switch_sample.json**

- **name:** Your Cisco Ethernet (Catalyst) Switch entity name.
- **displayName:** Your Cisco Ethernet (Catalyst) Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **omc_dispatch_url:** Under "value", following the string snmp://, provide the fully qualified hostname or IP address of the Cisco Ethernet (Catalyst) Switch.
- **omc_snmp_port:** Under "value", provide the port where the Cisco Ethernet (Catalyst) Switch listens for SNMP requests, 161 by default.
- **omc_snmp_timeout:** Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under "value", provide the SNMP version used to monitor the Cisco Ethernet (Catalyst) Switch.

 **Note:**

This is an optional property which is used only with SNMPV1Creds and allowed values are "1" or "2c". The default value is "2c".

Do not remove the square brackets.

Credential Files

Choose the creds json file according to what SNMP credentials you'd like to use - SNMP v2c or SNMP v3.

omc_cisco_eth_switch_snmpv2c_sample_creds.json

- **COMMUNITY:** SNMPv2c community string

omc_cisco_eth_switch_snmpv3_sample_creds.json

- **authUser:** SNMPv3 username
 - **authPwd:** password used for authentication
 - **authProtocol:** protocol used for authentication - supply either MD5 or SHA
 - **privPwd:** password used for encryption
-

Cisco Catalyst Ethernet Switch UI Fields

- **Entity Name:** Name of your Cisco Catalyst Ethernet Switch in Oracle Management Cloud
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Cisco Catalyst Ethernet Switch>
- **SNMP Port:** Port where Cisco Catalyst Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Cisco Catalyst Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** The SNMPv2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication.
 - **Authorization Protocol:** Protocol used for authentication (MD5 or SHA)
 - **Privacy Password:** Password used for encryption.
-

Cisco Nexus Ethernet Switch

Cisco Nexus Ethernet Switch JSON Files and Properties

Definition File: **omc_cisco_nexus_eth_switch_sample.json**

- **name:** Your Cisco Nexus Ethernet Switch entity name.
- **displayName:** Your Cisco Nexus Ethernet Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **omc_dispatch_url:** Under "value", following the string snmp://, provide the fully qualified hostname or IP address of the Cisco Nexus Ethernet Switch.
- **omc_snmp_port:** Under "value", provide the port where the Cisco Nexus Ethernet Switch listens for SNMP requests, 161 by default.
- **omc_snmp_timeout:** Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under "value", provide the SNMP version used to monitor the Cisco Nexus Ethernet Switch.

Note:

This is an optional property which is used only with SNMPV1Creds and allowed values are "1" or "2c". The default value is "2c".

Credential Files

omc_cisco_nexus_eth_switch_snmpv2_sample_creds.json

- **COMMUNITY:** Use this credential file if you have configured your switch with SNMPv1/v2.

omc_cisco_nexus_eth_switch_snmpv3_sample_creds.json

Under "value", within the square brackets, provide the SNMPv3 user name.

- **authPwd:** Under "value", within the square brackets, provide the auth password or empty out the field.
- **authProtocol:** Under "value", within the square brackets, provide the auth-method (SHA or MD5).
- **privPwd:** Under "value", within the square brackets, provide the priv method password, if priv is used. Only the DES priv method is supported.

Do not remove the square brackets.

Cisco Nexus Ethernet Switch UI Fields

- **Entity Name:** Name of your Cisco Nexus Ethernet Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Cisco Nexus Ethernet Switch>
- **SNMP Port:** Port where Cisco Nexus Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Cisco Nexus Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** SNMPv2c community string

SNMP V3

- **Username:** SNMPv3 username
- **Authorization Password:** Password used for authentication
- **Authorization Protocol:** Protocol used for authentication - supply either MD5 or SHA
- **Privacy Password:** Password used for encryption.

Oracle Power Distribution Unit (PDU)

Oracle Power Distribution Unit (PDU) Files and Properties

Definition File: `omc_oracle_pdu_sample.json`

- **name**: Your PDU entity name.
- **displayName**: Your PDU entity display name.
- **timezoneRegion**: Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: `America/New_York`
- **omc_dispatch_url**: Under "value", provide your PDU HTTP URL.
- **omc_snmp_port**: Under "value", provide your SNMP port, default is 161.
- **omc_snmp_timeout**: Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version**: Under "value", provide the SNMP version used to monitor PDU. Valid values are 1, 2 or 3.

Credential Files

You need to use HTTP credentials along with one of the SNMP credentials (v2c or v3)

HTTP credentials - part of `SNMPv1` and `SNMPv3 json`

- **username**: User name for HTTP communication.
- **password**: Password for user in HTTP communication.

SNMPv2c

`omc_oracle_pdu_sample_snmpv1_creds.json`

- **COMMUNITY**: Community String for SNMP communication

SNMP v3

`omc_oracle_pdu_sample_snmpv3_creds.json`

Use this credentials JSON file if using SNMP v3.

- **authUser**: Name of privileged user for SNMP communication
- **authPwd**: Password for privileged user for SNMP communication
- **authProtocol**: Encryption protocol to be used for SNMP communication
- **privPwd**: Password for SNMP communication

Do not remove the square brackets.

Arista Ethernet Switch

Arista Ethernet Switch JSON Files and Properties

Definition File: omc_arista_eth_switch_sample.json

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of Arista Ethernet Switch>
- **omc_snmp_port:** Port where Arista Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version:** SNMP version used to monitor Arista Ethernet Switch (2c or 3) - 2c by default (optional)

Credential Files

Choose the credential json file according to what SNMP credentials you'd like to use - SNMP v2c or SNMP v3. SNMP v2c

omc_arista_eth_switch_snmpv2_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2. Under "value", within the square brackets, provide the SNMPv2c community string used during the Arista Ethernet Switch configuration.

- **COMMUNITY:** SNMPv2c community string.

omc_arista_eth_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3

- **authUser:** Under "value", within the square brackets, provide provide SNMPv3 user name.
 - **authPwd:** Under "value", within the square brackets, provide the auth password or empty out the field.
 - **authProtocol:** Under "value", within the square brackets, provide the auth-method (SHA or MD5).
 - **privPwd:** Under "value", within the square brackets, provide the privilege method password, if privilege is used. Only the DES privilege method is supported.
-

Arista Ethernet Switch UI Fields

- **Entity Name:** Name of your Arista Ethernet Switch in Oracle Management Cloud
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Arista Ethernet Switch>
- **SNMP Port:** Port where Arista Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Arista Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **COMMUNITY:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication - supply either MD5 or SHA
 - **Privacy Password:** Password used for encryption.
-

Juniper Ethernet Switch

Juniper Ethernet Switch JSON Files and Properties

Credential File: **omc_juniper_eth_switch_sample.json**

- **name:** Your Juniper Switch entity name.
- **displayName:** Your Juniper Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under “value”, provide fully qualified host name or IP address of Juniper Switch
- **omc_dispatch_url:** Under “value”, following the string snmp://, provide the fully qualified hostname or IP address of Juniper Switch.
- **omc_snmp_port:** Under “value”, provide your SNMP port, default is 161.
- **omc_snmp_timeout:** Under “value”, provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under “value”, provide the SNMP version used to monitor Juniper Switch, 2c by default.

Credential Files

omc_juniper_eth_switch_snmpv2c_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2.

- **COMMUNITY:** Use this credential file if you have configured your switch with SNMPv1/v2.

omc_juniper_eth_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3

- **authUser:** Use this credential file if you have configured your switch with SNMPv3. Under “value”, within the square brackets, provide provide SNMPv3 user name.
- **authPwd:** Under “value”, within the square brackets, provide the auth password or empty out the field.
- **authProtocol:** Under “value”, within the square brackets, provide the auth-method (SHA or MD5).
- **privPwd:** Under “value”, within the square brackets, provide the priv method password, if priv is used. Only the DES priv method is supported.

Do not remove the square brackets.

Juniper Ethernet Switch UI Fields

- **Entity Name:** Name of your Juniper Ethernet Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Juniper Ethernet Switch>
- **SNMP Port:** Port where Juniper Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring Juniper Ethernet Switch.

Monitoring Credentials

Choose the credential JSON file according to what SNMP credentials you'd like to use - SNMP v2c or SNMP v3.

SNMP V1/V2

- **Community String:** SNMPv2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

Oracle Infiniband Switch

Oracle Infiniband Switch JSON Files and Properties

Definition File: **omc_oracle_ib_switch_sample.json**

- **name:** Your InfiniBand Switch entity name.
- **displayName:** Your InfiniBand Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under “value”, provide fully qualified host name or IP address of InfiniBand Switch
- **omc_dispatch_url:** Under “value”, following the string snmp://, provide the fully qualified hostname or IP address of InfiniBand Switch.
- **omc_snmp_port:** Under “value”, provide your SNMP port, default is 161.
- **omc_snmp_timeout:** Under “value”, provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under “value”, provide the SNMP version used to monitor InfiniBand Switch, 2c by default.

Credential Files

omc_oracle_ib_switch_snmpv2c_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2.

- **COMMUNITY:** Under “value”, within the square brackets, provide the SNMPv2c community string used during the InfiniBand Switch configuration.

omc_oracle_ib_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3

- **authUser:** Under “value”, within the square brackets, provide provide SNMPv3 user name.
- **authPwd:** Under “value”, within the square brackets, provide the auth password or empty out the field.
- **authProtocol:** Under “value”, within the square brackets, provide the auth-method (SHA or MD5).
- **privPwd:** Under “value”, within the square brackets, provide the priv method password, if priv is used.

Do not remove the square brackets.

Oracle Infiniband Switch UI Fields

- **Entity Name:** Name of your Oracle Infiniband Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Oracle InfiniBand Switch>
- **SNMP Port:** Port where Oracle InfiniBand Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring your Oracle Infiniband Switch.

Monitoring Credentials

SNMP V1/V2

- **Community String:** SNMPv2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

Oracle Fabric Manager / Virtual Networking / Xsigo

Oracle Fabric Manager / Virtual Networking / Xsigo JSON Files and Properties

 **Note:**

ONLY proper SSL certificates of OVN/OFM are supported. For self-signed certificates, manual addition to the agent keystore is required. To manually add a self-signed certificate to the agent keystore, run the following command:

```
omcli secure add_trust_cert_to_jks -password
<ask_oracle_support> -trust_certs_loc </path/to/
certificateOfOFMServer.crt> -alias
<hostname_of_OFM>
```

omc_oracle_ovn_sample.json

- **name:** Your OFM/OVN entity name.
- **displayName:** Your OFM/OVN entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **omc_ovn_hostname:** Under “value”, provide fully qualified host name or IP address of the OFM/OVM host.
- **omc_ovn_rest_port:** Under “value”, provide your OFM/OVN REST port, default is 8443.

Credential File: **omc_oracle_ovn_sample_creds.json**

- **username:** Under “value”, provide fully qualified host name or IP address of the OFM/OVM host.
- **password:** Under “value”, within the square brackets, provide the OFM/OVN user password.

Do not remove the square brackets.

Oracle Fabric Manager / Virtual Networking / Xsigo UI Fields

- **Entity Name:** Your OFM/OVN entity name.
- **Host Name:** The fully qualified host name or IP address of the OFM/OVM host.
- **REST Port:** The OFM/OVN REST port, default is 8443.
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where OFM/OVM is installed.

Monitoring Credentials

- **Username:** The OFM/OVN user.
 - **Password:** The OFM/OVM user password.
-

Brocade Fiber Channel Switch

Brocade Fiber Channel Switch JSON Files and Properties

Definition File: **omc_brocade_fc_switch_sample.json**

- **name:** Your Brocade Fibre Channel Switch entity name.
- **displayName:** Your Brocade Fibre Channel Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under “value”, provide fully qualified host name or IP address of the Brocade Fibre Channel Switch
- **omc_dispatch_url:** Under “value”, following the string snmp://, provide the fully qualified hostname or IP address of the Brocade Fibre Channel Switch.
- **omc_snmp_port:** Under “value”, provide the port where the Brocade Fibre Channel Switch listens for SNMP requests. The default is 161.
- **omc_snmp_timeout:** Under “value”, provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under “value”, provide the SNMP version used to monitor the Brocade Fibre Channel Switch.

Credential Files

omc_brocade_fc_switch_snmpv1_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2.

- **community:** Under “value”, within the square brackets, provide the SNMPv2c community string used during the Brocade Fibre Channel Switch configuration.

omc_brocade_fc_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3.

- **authUser:** Under “value”, within the square brackets, provide the SNMPv3 username.
 - **authPwd:** Under “value”, within the square brackets, provide the authorization password or empty out the field. .
 - **authProtocol:** Under “value”, within the square brackets, provide the authorization method (SHA or MD5).
 - **privPwd:** Under “value”, within the square brackets, provide the privilege method password, if privilege is used. Only the DES privilege method is supported.
-

Brocade Fiber Channel Switch UI Fields

- **Entity Name:** Name of your Brocade Fiber Channel Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Brocade Fiber Channel Switch>
- **SNMP Port:** Port where Brocade Fiber Channel Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring your Brocade Fiber Channel Switch.

Monitoring Credentials

SNMP V1/V2:

- **Community String:** SNMPv1/v2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

SCOM (System Center Operations Manager)

SCOM (System Center Operations Manager) JSON Files and Properties

Definition File: `omc_microsoft_scom_example.json`

- **name:** Your SCOM entity name.
- **displayName:** Your SCOM entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: `America/New_York`
- **omc_sdk_host:** Fully qualified host name or IP address of the server which hosts SCOM.

Credential File: `omc_microsoft_scom_creds.json`

- **username:** Username of the account which has access to SCOM..
 - **password:** Password of the account which has access to SCOM.
 - **domain:** Windows domain of the account which has access to SCOM.
-

SCOM (System Center Operations Manager) UI Fields

- **Entity Name:** Your SCOM entity name in Oracle Management Cloud.
- **SCOM SDK Host:** Host name or IP address of an SCOM SDK Host.
- **Cloud Agent:** Cloud agent monitoring the host where SCOM is installed.

Monitoring Credentials (SCOM Credentials)

- **Username:** Username of the account which has access to SCOM..
 - **Password:** Password of the account which has access to SCOM.
 - **Domain:** Windows domain of the account which has access to SCOM.
-

Juniper SRX Firewall

Juniper SRX Firewall JSON Files and Properties

omc_juniper_srx_sample.json

- **name:** Your Juniper SRX Firewall entity name.
- **displayName:** Your Juniper SRX Firewall entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under “value”, provide fully qualified host name or IP address of the Juniper SRX Firewall.
- **omc_dispatch_url:** Under “value”, following the string snmp://, provide the fully qualified hostname or IP address of the Juniper SRX Firewall.
- **omc_snmp_port:** Under “value”, provide the port where the Juniper SRX Firewall listens for SNMP requests. The default is 161.
- **omc_snmp_timeout:** Under “value”, provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under “value”, provide the SNMP version used to monitor the Juniper SRX Firewall.

Credential Files

omc_juniper_srx_snmpv2_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2.

- **community:** Under “value”, within the square brackets, provide the SNMPv2c community string used during the Juniper SRX Firewall configuration.

omc_juniper_srx_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3.

- **authUser:** Under “value”, within the square brackets, provide the SNMPv3 username.
 - **authPwd:** Under “value”, within the square brackets, provide the authorization password or empty out the field. .
 - **authProtocol:** Under “value”, within the square brackets, provide the authorization method (SHA or MD5).
 - **privPwd:** Under “value”, within the square brackets, provide the privilege method password, if privilege is used. Only the DES privilege method is supported.
-

Juniper SRX Firewall UI Fields

- **Entity Name:** Name of your Juniper SRX Firewall in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Juniper SRX Firewall>
- **SNMP Port:** Port where Juniper SRX Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring your Juniper SRX Firewall.

Monitoring Credentials

SNMP V1/V2:

- **Community String:** SNMPv1/v2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

Oracle SOA Infrastructure

Oracle SOA Infrastructure JSON Files and Properties

Oracle SOA Infrastructure entities are automatically discovered as part of the WebLogic Domain discovery.

 **Note:**

When you add a WebLogic Domain entity (requiring credentials), because Oracle Management Cloud connects to the WebLogic Admin Server, all WebLogic Clusters and WebLogic Servers that are part of that domain are automatically discovered. There's no need to add them separately.

Oracle Service Bus

Oracle Service Bus JSON Files and Properties

Definition File: **omc_oracle_servicebus_sample.json**

- **name:** Your WebLogic Domain name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **port:** Under "value", provide the port used for WebLogic Admin Server(Console).
- **protocol:** Under "value", provide the protocol used for WebLogic Server - For example: t3
- **admin_server_host:** Under "value", provide the fully-qualified host name where the WebLogic Admin Server is installed.

Credential File: **omc_oracle_servicebus_sample_creds.json**

- **username:** Under "value", within the square brackets, provide the WebLogic Domain monitoring user name.
 - **password:** Under "value", within the square brackets, provide the WebLogic Domain monitoring user's password.
-

Multi-Entity

Multi-Entity JSON Files and Properties

Use the multi-entity JSON files to add multiple entities of various types at the same time. These JSON files include entries for all entities. Edit all field values for the entities you are adding and remove all other entries in the files that you aren't using. See more details in the Oracle by Example (OBE) tutorial on [Adding Multiple Entities With a Single JSON File](#)

Definition File: **omc_add_multi_entities.json**

Credential File: **omc_add_multi_entities_creds.json**

Fujitsu Computers

Fujitsu Computers JSON Files and Properties

Definition File: **omc_fujitsu_server_sample.json**

- **omc_dispatch_url**: snmp://<Fully qualified host name or IP address of Fujitsu Server>
- **omc_snmp_port**: Port where Fujitsu Server listens for SNMP requests (default 161)
- **omg_snmp_timeout**: Timeout for SNMP requests in seconds (default 20)
- **omc_snmp_version**: SNMP version used to monitor Fujitsu Server (2c or 3) (default 3)

Credential File: **omc_fujitsu_server_creds_sample.json**

SNMP v2c

- **COMMUNITY**: SNMPv2c community string

SNMP v3

- **authUser**: SNMPv3 username.
 - **authPwd**: Password used for authentication.
 - **authProtocol**: protocol used for authentication - supply MD5
 - **privPwd**: password used for encryption
 - **privProtocol**: Protocol used for encryption - supply DES
-

Fujitsu Computer UI Fields

- **Entity Name**: Name of your Fujitsu Server in Oracle Management Cloud.
- **Dispatch URL**: snmp://<Fully qualified host name or IP address of the Fujitsu Server.>
- **SNMP Port**: Port where the Fujitsu Server listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent**: Cloud agent monitoring your Fujitsu Server.

Monitoring Credentials

SNMP V1/V2:

- **Community String**: SNMPv1/v2c community string.

SNMP V3

- **Username**: SNMPv3 username.
 - **Authorization Password**: Password used for authentication
 - **Authorization Protocol**: Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password**: password used for encryption
-

Intel/SPARC Computers

Intel/SPARC Computers JSON Files and Properties

Definition File: **omc_ilom_server.json**

- **omc_dispatch_url**: ilom-ssh://<fully qualified host name or IP address of ILOM Server>

Credential File: **omc_ilom_server_creds.json**

- **username**: ILOM Server username (use Administrator role).
 - **password**: ILOM Server password (use Administrator role).
-

VMware vCenter

VMware vCenter JSON Files and Properties

Definition File: **omc_vmware_vcenter_sample.json**

- **omc_virtual_mgmt_system_id:** VMware vCenter Server Instance UUID
- **omc_virtual_type:** VMware
- **omc_dispatch_url:** vmware-https://<Fully qualified host name or IP address of vCenter>/sdk/vimservice

Credential File: **omc_vmware_vcenter_sample_creds.json**

- **username:** VMware vCenter username (use Administrator role).
 - **password:** VMware vCenter password (use Administrator role).
-

NGINX

NGINX Files and Properties

Definition File: **omc_nginx.json**

- **host_name:** Host Name of the Nginx Target
 - **listen_port:** Nginx Server Port Number for connection to Nginx Status page
 - **install_home:** Nginx Server install directory
-

NGINX UI Fields

- **Entity Name:** Name of your NGINX entity in Oracle Management Cloud.
 - **Host Name:** Host where the NGINX server is running.
 - **Nginx Listen Port:** NGINX Server Port Number for connection to NGINX Status page.
 - **Nginx Binary File Path:** Full path to the NGINX binary file.
 - **Nginx PID File Path:** Full path to the NGINX PID file.
 - **Nginx Status Page URL:** URL used to access the NGINX status page.
 - **Cloud Agent:** Cloud agent monitoring the host where the NGINX server is installed.
-

Apache SOLR

Apache SOLR JSON Files and Properties

Definition Files:

omc_solr_instance_credless.json

omc_solr_instance_creds.json

omc_solrcloud_credless.json

omc_solrcloud_creds.json

Credential Files:

solr_basic_authentication.json

solr_client_authentication.json

solr_client_with_basic_authentication.json

Replace any text inside brackets <> excluding these brackets with your values according the legend inside <>

Notes:

1. Credential JSONs are same for both standalone (omc_solr_instance) & solrcloud (omc_solrcloud)
2. For secure solr standalone & solrcloud (with or without creds), server certificates for all the instances need to be added to the agent keystore using following command `omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc <certificate location> -alias <alias>`
3. For client authentication the type of the agent keystore is JKS keystore

Examples of Base URLs:

`non secure – http://<hostname>:<port>/solr/`

`secure — https://<hostname>:<port>/solr/`

Example of credentialRefs in the entity JSONs:

`"credentialRefs" : ["SolrKeyStoreCredRef"] ---- for client authentication`

`"credentialRefs" : ["SolrBasicCredRef"] ---- for basic authentication`

`"credentialRefs" : ["SolrBasicCredRef", "SolrKeyStoreCredRef"] ---- for client with basic authentication`

Apache SOLR UI Fields

- **Discover Using Credentials:** Discover Apache SOLR using Apache SOLR credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Base URL:** The base URL for REST API invocations.
- **Cloud Agent:** Cloud agent monitoring the host where the Apache SOLR is installed.

Monitoring Credentials

Basic Authentication

- **Username:** Apache SOLR username.
- **Password:** Apache SOLR password.

Client Authentication

- **Keystore Location:** Full path to the location of the keystore file.
- **Keystore Password:** Keystore authentication password.

BasicClient

Basic Authentication

- **Username:** Apache SOLR username.
- **Password:** Apache SOLR password.

Client Authentication

- **Keystore Location:** full path to the location of the keystore file.
 - **Keystore Password:** Keystore authentication password.
-

Hadoop

Hadoop JSON Files and Properties

 **Note:**

Hadoop uses primary and secondary nodes to maintain availability. If one node goes down, the secondary node is used to fetch information. For both nodes, the primary and secondary roles can be switched at any given time. For this reason, you specify two nodes and two Resource Manager URLs without specifying whether they are primary or secondary.

Definition File: **hadoop_credless.json**

Use this file if Hadoop was configured with no credentials.

- **Name:** Hadoop entity name.
- **displayName:** Hadoop entity display name.
- **omc_nn1_metric_url** (Name Node 1): Name Node 1 URL with port.
- **omc_nn2_metric_url** (Name Node 2): Name Node 2 URL with port.
- **omc_rm1_metric_url** (Resource Manager 1): Resource Manager Node 1 URL with port.
- **omc_rm2_metric_url** (Resource Manager 2): ResourceManager Node 2 URL with port.

Example URL

```
http://<HOSTNAME>:<PORT>/ (Rest API URL for
    Invocation)
```

Definition File: **hadoop_creds.json**

- **name:** Hadoop entity name.
- **displayName:** Hadoop entity display name.
- **credentialRefs:** Hadoop credential information (hadoopTrustStore and hadoopSPNEGOCredentials) defined in the hadoop_credentials_input.json file.
- **omc_nn1_metric_url:** (Name Node 1): Name Node 1 URL with port.
- **omc_nn2_metric_url:** (Name Node 2): Name Node 2 URL with port.
- **omc_rm1_metric_url:** (Resource Manager 1): Resource Manager Node 1 URL with port.
- **omc_rm2_metric_url:** (Resource Manager 2): ResourceManager Node 2 URL with port.

Example URL

```
http://<HOSTNAME>:<PORT>/ (Rest API URL for Invocation)
```

Credential File: **hadoop_credentials_input.json**

- **hadoopTrustStore** consists of the following user-defined properties:
 - *StoreLocation:* Path of Truststore file.
 - *StorePassword:* Truststore Password
 - **hadoopSPNEGOCredentials** consists of the following user-defined properties:
 - *Alias:* Alias name.
 - *Password:* Alias password.
 - *KRB5Conf:* Path of krb5.conf file.
-

Hadoop UI Fields

- **Discover Using Credentials:** Discover Hadoop using Hadoop credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Metric URL for NameNode 1:** Name Node 1 URL with port..
- **Metric URL for NameNode 2:** Name Node 2 URL with port.
- **Metric URL for Resource Manager 1:** Resource Manager Node 1 URL with port.
- **Metric URL for Resource Manager 2:** ResourceManager Node 2 URL with port.
- **Cloud Agent:** Cloud agent monitoring the host where Hadoop is installed.

Monitoring Credentials

SSL Trust Store

- **Store Location:** Path of the Truststore file.
- **Store Password:** The keystore password to access the JKS file.
- **Store Type:** Currently, only JKS is supported.

Alias

- **Alias Name:** Your alias name.
 - **Password:** Your alias password.
 - **Path of krb5.conf file:** Full path to the krb5.conf file.
-

Arbor TMS Firewall

Arbor TMS Firewall JSON Files and Properties

Definition File: **omc_arbor_tms_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of the Arbor TMS Firewall>
- **omc_snmp_port:** Port where the Arbor TMS Firewall listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version:** SNMP version used to monitor the Arbor TMS Firewall (2c or 3) - 2c by default (optional)

Credential Files

omc_arbor_tms_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_arbor_tms_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication (MD5 or SHA).
 - **privPwd:** Password used for encryption.
-

Arbor TMS Firewall UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of the Arbor TMS Firewall>
- **SNMP Port:** Port where the Arbor TMS Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Arbor TMS Firewall is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

Arbor CP Firewall

Arbor CP Firewall JSON Files and Properties

Definition File: **omc_arbor_cp_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of the Arbor CP Firewall>
- **omc_snmp_port:** Port where the Arbor CP Firewall listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version:** SNMP version used to monitor the Arbor CP Firewall (2c or 3) - 2c by default (optional)

Credential Files

omc_arbor_cp_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_arbor_cp_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication (MD5 or SHA).
 - **privPwd:** Password used for encryption.
-

Arbor CP Firewall UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of the Arbor CP Firewall>
- **SNMP Port:** Port where the Arbor CP Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Arbor CP Firewall is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

Juniper Netscreen Firewall

Juniper Netscreen Firewall JSON Files and Properties

Definition File: **omc_juniper_netscreen_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of Juniper Netscreen Firewall>
- **omc_snmp_port:** Port where Juniper Netscreen Firewall listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version:** SNMP version used to monitor Juniper Netscreen Firewall (2c or 3) - 2c by default (optional)

Credential Files

omc_juniper_netscreen_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_juniper_netscreen_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username
 - **authPwd:** Password used for authentication
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA
 - **privPwd:** password used for encryption
-

Juniper Netscreen Firewall UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Juniper Netscreen Firewall>
- **SNMP Port:** Port where Juniper Netscreen Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Juniper Netscreen Firewall is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

Juniper MX Router

Juniper MX Router JSON Files and Properties

Definition File: **omc_juniper_mx_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of Juniper MX Router>
- **omc_snmp_port:** Port where Juniper MX Router listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version:** SNMP version used to monitor Juniper MX Router (2c or 3) - 2c by default (optional)

Credential Files

omc_juniper_mx_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_juniper_mx_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd:** Password used for encryption.
-

Juniper MX Router UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Juniper MX Router>
- **SNMP Port:** Port where Juniper MX Router listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where the Juniper MX Router is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

F5 BIG-IP LTM

F5 BIG-IP LTM JSON Files and Properties

Definition File: **omc_f5_bigip_ltm_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of F5 BIG-IP LTM>
- **omc_snmp_port:** Port where F5 BIG-IP LTM listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version:** SNMP version used to monitor F5 BIG-IP LTM (2c or 3) - 2c by default (optional)

Credential Files

omc_f5_bigip_ltm_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_f5_bigip_ltm_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd:** Password used for encryption.
-

F5 BIG-IP LTM UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of F5 BIG-IP LTM>
- **SNMP Port:** Port where F5 BIG-IP LTM listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where F5 BIG-IP LTM is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

F5 BIG-IP DNS

F5 BIG-IP DNS JSON Files and Properties

Definition File: **omc_f5_bigip_dns_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of F5 BIG-IP DNS>
- **omc_snmp_port:** Port where F5 BIG-IP DNS listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version:** SNMP version used to monitor F5 BIG-IP DNS (2c or 3) - 2c by default (optional)

Credential Files

omc_f5_bigip_dns_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_f5_bigip_dns_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd:** Password used for encryption.
-

F5 BIG-IP DNS UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of F5 BIG-IP DNS>
- **SNMP Port:** Port where F5 BIG-IP DNS listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where F5 BIG-IP DNS is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

NetApp FAS

NetApp FAS JSON Files and Properties

Definition File: **omc_netapp_fas_sample.json**

- **host_name:** Fully qualified domain name (FQDN) or IP of the NetApp storage.
- **omc_snmp_port:** Port to use for SNMP communication with NetApp storage.
- **omc_snmp_timeout:** Timeout for SNMP communication with NetApp storage.
- **omc_snmp_version:** Version of SNMP protocol to use for communication with NetApp storage.
- **omc_snmp_community:** SNMP community string to use for communication with NetApp storage.

Credential File: **omc_netapp_fas_snmp_sample_creds.json**

- **authUser:** Name of a privileged user for SNMP communication.
 - **authPwd:** Password for a privileged user for SNMP communication.
 - **authProtocol:** Encryption protocol to be used for SNMP communication.
 - **privPwd:** Password for SNMP communication.
-

NetApp FAS UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
 - **Host Name:** Fully qualified domain name (FQDN) or IP of the NetApp storage.
 - **SNMP Port:** Port to use for SNMP communication with NetApp storage.
 - **SNMP Timeout:** Timeout for SNMP communication with NetApp storage.
 - **SNMP Version:** Version of SNMP protocol to use for communication with NetApp storage.
 - **SNMP Community:** SNMP community string to use for communication with NetApp storage.
 - **Cloud Agent:** Cloud agent monitoring the host where NetApp FAS is installed.
-

ZFS Storage Appliance

ZFS Storage Appliance JSON Files and Properties

Definition File: **omc_oracle_zfs_storage_appliance_sample.json**

- **omc_zfssa_hostname**: ONLY IP of the ZFS Storage Appliance (if you use hostname/fully qualified domain name, you will trigger a REST fetchlet problem with certificate validation: (javax.net.ssl.SSLProtocolException: handshake alert: unrecognized_name))
- **omc_zfssa_port**: Port to use for REST API communication with ZFS Storage Appliance storage
- **omc_ssl_trust_server_cert**: Flag indicating whether to trust self-signed certificates.

Credential File: **omc_oracle_zfs_storage_appliance_sample_creds.json**

- **Alias**: Alias (username/login name) to be used for the ZFS Storage Appliance REST API
 - **Password**: Password for the ZFS Storage Appliance REST API alias.
-

ZFS Storage Appliance UI Fields

- **Entity Name**: Name of this entity displayed in the Oracle Management Cloud console.
- **ZFFSA IP Address**: IP address of the ZFS storage appliance with REST API.
- **ZFFSA Port**: Port of the storage appliance REST API.
- **Trust Any Server Certificate**: False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent**: Cloud agent monitoring the host where the ZFS Storage Appliance is installed.

Monitoring Credentials (ZFFSA Credentials)

- **Username**: Storage appliance username.
 - **Password**: Storage appliance password.
-

Kubernetes Cluster

Kubernetes Cluster JSON Files and Properties

Replace any text inside brackets <> excluding these brackets with your values according the legend within the brackets <>.

See [Kubernetes Cluster](#) for property descriptions.

Definition Files

omc_kubernetes_cluster_insecure.json

omc_kubernetes_cluster_secure.json

Credential Files

omc_kubernetes_cluster_basic_creds.json

omc_kubernetes_cluster_keystore_creds.json

omc_kubernetes_cluster_token_creds.json

Kubernetes Cluster UI Fields

- **Discover Using Credentials:** Discover Kubernetes Cluster using Kubernetes Cluster credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Kubernetes Master URL:** Base URL of the API Server on the Kubernetes Master Node. The URL is of the form `http(s)://<hostname>:<port>`
- **Host Name:** Hostname of the Kubernetes master node
- **Heapster URL:** Base URL of Heapster. This needs to be specified if the performance metrics are to be collected from Heapster. If heapster is running inside Kubernetes as a cluster service the Base URL is of the form `http(s)://<host>:<port>/api/v1/namespaces/kube-system/services/heapster/proxy` Here, the host & port are same as in `omc_kubernetes_master_url`
- **Cloud Agent:** Cloud agent monitoring the host where the Kubernetes Cluster is installed.

Monitoring Credentials

Token Credentials

- **Token:** Token of the user going to discover Kubernetes
- **Keystore Certificate:** Certificate of Kubernetes API Server on Master Node. Users need to specify the text inside the certificate file if added from UI. In `omcli`, users need to create a Java Keystore, add certificate to that and specify the file path.
- **Certificate Alias:** Alias for the Certificate. This should be unique alphanumeric string
- **Trust Store Password:** Password of agent's Trust Store. This password is "welcome"

Basic Credentials

- **Username:** Username of the user going to discover Kubernetes
- **Password:** Password used for authentication.
- **Keystore Certificate:** Certificate of Kubernetes API Server on Master Node. Users need to specify the text inside the certificate file if added from UI. In `omcli`, users need to create a Java Keystore, add certificate to that and specify the file path.
- **Certificate Alias:** Alias for the Certificate. This should be unique alphanumeric string
- **Trust Store Password:** Password of agent's Trust Store. This password is "welcome"

Keystore Credentials

- **Store Location:** Location of Client keystore. This Java Keystore file (JKS) should contain client's certificate.
 - **Store Type:** Store type. This value is always set to "JKS"
 - **Store Password:** The keystore password to access the JKS file.
 - **Keystore Certificate:** Certificate of Kubernetes API Server on Master Node. Users need to specify the text inside the certificate file if added from UI. In `omcli`, users need to create a Java Keystore, add certificate to that and specify the file path.
 - **Certificate Alias:** Alias for the Certificate. This should be unique alphanumeric string
 - **Trust Store Password:** Password of agent's Trust Store. This password is "welcome"
-

Oracle ES2 Ethernet Switch

Oracle ES2 Ethernet Switch JSON Files and Properties

Definition File: **omc_es2_sample.json**

- **omc_dispatch_url**: snmp://<Fully qualified host name or IP address of Oracle ES2 Ethernet Switch>
- **omc_snmp_port**: Port where Oracle ES-2 Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version**: SNMP version used to monitor Oracle ES-2 Ethernet Switch (2c or 3) - 2c by default (optional)

Credential Files

Choose the credential JSON file according to the SNMP version credentials you're using (SNMP v2c or SNMP v3).

omc_es2_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY**: SNMPv2c community string.

omc_es2_snmpv3_sample_creds.json

SNMP v3

- **authUser**: SNMPv3 username.
 - **authPwd**: Password used for authentication.
 - **authProtocol**: Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd**: Password used for encryption.
-

Oracle ES2 Ethernet Switch UI Fields

- **Entity Name**: Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL**: snmp://<Fully qualified host name or IP address of Oracle ES2 Ethernet Switch>
- **SNMP Port**: Port where Oracle ES-2 Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent**: Cloud agent monitoring the host where the Oracle ES2 Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String**: Community String for SNMP communication

SNMP V3

- **Username**: SNMPv3 username.
 - **Authorization Password**: Password used for authentication
 - **Authorization Protocol**: Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password**: password used for encryption
-

Oracle Flash Storage

Oracle Flash Storage JSON Files and Properties

 **Note:**

Self-signed Certificate Limited Support: ONLY proper SSL certificates of Oracle Flash Storage are supported out-of-box. Self-signed certificates need to be added manually to agent keystore using the following command:

```
omcli secure add_trust_cert_to_jks -password
<ask_oracle_support> -trust_certs_loc </path/to/
certificateOfFS.crt> -alias
<ideally_hostname_of_FS>
```

Definition File: **omc_oracle_flash_storage_sample.json**

- **omc_oracle_flash_storage_hostname:** Fully qualified domain name (FQDN) or IP of the Oracle Flash Storage.
- **omc_oracle_flash_storage_port:** Port to use for REST API communication with Oracle Flash Storage.
- **omc_oracle_flash_storage_ssl_trust_server_cert:** Name of the certificate for REST API communication with Oracle Flash Storage.

Credential File: **omc_oracle_flash_storage_creds_sample.json**

- **Alias:** Alias (username) to be used for Oracle Flash Storage REST API
 - **Password:** Password for alias for Oracle Flash Storage REST API
-

Apache Cassandra DB

Apache Cassandra DB JSON Files and Properties

Definition File: **omc_cassandra_db.json**

- **displayName:** This is Apache Cassandra Database Entity Display Name which is displayed in the Oracle Infrastructure Monitoring UI
- **timezoneRegion:** Time Zone Example: PDT, GMT
- **omc_url:** connection url to connect to the installed Apache Cassandra database; host:port
- **host_name :** Fully-qualified Host Name where Cassandra database is installed.
- **omc_port :** Apache Cassandra Database port.
- **cassandra_home:** Location of the Cassandra Installation directory.

Credential File: **omc_cassandra_db_creds.json**

- **DBUserName:** Cassandra DB username.
 - **DBPassword:** Cassandra DB user password.
-

Apache Cassandra DB UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where Cassandra database is installed.
- **Port:** Apache Cassandra Database port.
- **Cloud Agent:** Cloud agent monitoring the host where the Apache Cassandra DB is installed.

Monitoring Credentials (Cassandra JMX Credentials)

- **Username:** Cassandra DB username.
 - **Password:** Password used for authentication.
-

EMC VMAX Storage

EMC VMAX Storage JSON Files and Properties

Definition File: **omc_emc_vmax_sample.json**

- **symmetrix_id:** Symmetrix ID for the Storage Array (storage array identifier)
- **omc_emc_vmax_univmax_hostname:** Fully qualified domain name or IP of EMC VMAX Unisphere for storage.
- **omc_emc_vmax_univmax_port:** Port to use for REST API communication with EMC VMAX Unisphere for storage.
- **omc_emc_vmax_ssl_trust_server_cert:** Name of the certificate for REST API communication with EMC VMAX Unisphere for storage.

Credential File: **omc_emc_vmax_creds_sample.json**

- **Alias:** Alias (username) to be used for EMC VMAX REST API.
 - **Password:** Password for the EMC VMAX REST API alias.
-

EMC VNX Storage

EMC VNX Storage JSON Files and Properties

Definition File: **omc_emc_vnx_instance_sample.json**

- **omc_emc_vnx_block_storage_binary:** Path to the navseccli binary on the remote host (including binary name).
- **omc_emc_vnx_binary_hostname_url:** SSH connection string such as "ssh://hostname", to the remote host where navseccli is installed.
- **omc_emc_vnx_binary_hostname_tmp:** Location for temporary files on the remote host where navseccli is installed (including trailing path separator). Example: "/tmp/"
- **omc_emc_vnx_storage_processor_a:** IP address of Storage Processor A for a discovered storage array.
- **omc_emc_vnx_storage_processor_b:** IP address of Storage Processor B for a discovered storage array.

Credential File: **omc_emc_vnx_creds_sample.json**

- **username:** Username used for an SSH connection to a remote host where navseccli is installed,
 - **userpass:** Password of the user making an SSH connection to the remote host where navseccli is installed,
 - **cmdusername:** Username of the user running the navsecclicommand.
 - **cmduserpass:** Password for the user running the navsecclicommand,
 - **cmduserscope:** Scope of the user for the command: 0,1,2 (global, local,, LDAP).
-

EMC VNX Storage UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **NaviSecCli binary path:** Path to the navseccli binary on the remote host (including binary name).
- **NaviSecCli hostname URL:** SSH connection string such as "ssh://hostname", to the remote host where navseccli is installed.
- **Storage Processor A:** IP address of Storage Processor A for a discovered storage array.
- **Storage Processor B:** IP address of Storage Processor B for a discovered storage array.
- **NaviSecCli Temp Files:** Location for temporary files on the remote host where naviseccli is installed (including trailing path separator). Example: "/tmp/"
- **Cloud Agent:** Cloud agent monitoring the host where EMC VNX Storage is installed.

Monitoring Credentials

- **Username:** Username used for an SSH connection to a remote host where naviseccli is installed,
 - **Password:** Password of the user making an SSH connection to the remote host where naviseccli is installed,
 - **CMD Username:** Username of the user running the navisecclicommand.
 - **CMD Password:** Password for the user running the navisecclicommand,
 - **User Scope:** Scope of the user for the command: 0,1,2 (global, local,, LDAP).
-

Oracle VM Server for SPARC (LDoms)

Oracle VM Server for SPARC (LDoms) JSON Files and Properties

Definition File: **omc_sparc_ldoms_sample.json**

- **omc_virtual_platform_id:** LDoms Control Domain UUID
 - **omc_virtual_type:** LDoms
 - **omc_dispatch_url:** local://localhost
-

Apache Zookeeper

Apache Zookeeper JSON Files and Properties

Definition File: **omc_apache_zookeeper.json**

- **listen_port:** Zookeeper port used to listen for client connections.
 - **host_name:** Zookeeper host name.
-

Apache Zookeeper UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
 - **Host Name:** Zookeeper host name.
 - **Zookeeper listening port:** Zookeeper port used to listen for client connections.
 - **Cloud Agent:** Cloud agent monitoring the host where Apache Zookeeper is installed.
-

L2/L3 Generic Network Node

L2/L3 Generic Network Node JSON Files and Properties

Definition File: **omc_network_node_sample.json**

- **omc_node_id**: The string to be used as the identifying property for the hardware.
- **omc_hw_vendor**: The vendor name that will be exposed in the Product metric group
- **omc_hw_product** : The product name that will be exposed in the Product metric group
- **omc_hw_version** : The product version that will be exposed in the Product metric group
- **omc_dispatch_url**: snmp://<Fully qualified host name or IP address of network node>
- **omc_snmp_port**: Port where network hardware listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version**: SNMP version used to monitor the network node (2c or 3) - 2c by default (optional)

Credential Files

Choose the creds JSON file according to what SNMP credentials you'd like to use - SNMP v2c or SNMP v3.

omc_network_node_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY**: SNMPv2c community string

omc_network_node_snmpv3_sample_creds.json

SNMP v3

- **authUser**: SNMPv3 username
 - **authPwd**: Password used for authentication
 - **authProtocol**: Protocol used for authentication - supply either MD5 or SHA
 - **privPwd**: Password used for encryption
-

L2/L3 Generic Network Node UI Fields

- **Entity Name**: Name of this entity displayed in the Oracle Management Cloud console.
- **Network Node ID**: The string to be used as the identifying property for the hardware.
- **Vendor**: The vendor name that will be exposed in the Product metric group
- **Product** : The product name that will be exposed in the Product metric group
- **Version**: The product version that will be exposed in the Product metric group
- **Dispatch URL**: snmp://<Fully qualified host name or IP address of network node>
- **SNMP Port**: Port where network hardware listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent**: Cloud agent monitoring the host where the L2/L3 Generic Network Node is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String**: Community String for SNMP communication

SNMP V3

- **Username**: SNMPv3 username.
 - **Authorization Password**: Password used for authentication
 - **Authorization Protocol**: Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password**: password used for encryption
-

JBoss Server/Domain

JBoss Server/Domain JSON Files and Properties

Definition Files

omc_jboss_standalone_j2eeserver_sample.json

omc_jboss_standalone_j2eeserver_secure_sample.json

Credential Files

omc_jboss_standalone_j2eeserver_sample_creds.json

omc_jboss_standalone_j2eeserver_secure_sample_creds.json

JBoss Standalone Server:

- **host_name**: Your Fully-qualified JBoss Standalone J2EE Server Host Name
- **omc_management_port**: Your JBoss Management Console Port

For Non-Secure (no-SSL):

- CredType:MonitorCreds

Properties:

- **user_name**: Your JBoss Management User Name
- **password**: Your JBoss Management User Password

For Secure (SSL):

- CredType:MonitorCreds

Properties:

- **user_name**: Your JBoss Management User Name
- **password**: Your JBoss Management User Password
- **ssl_trust_store**: Your OMC Cloud Agent Truststore Location
- **ssl_trust_store_password**: Your OMC Cloud Agent Truststore Password

JBoss Domain:

- **omc_host_name**: Your Fully-qualified JBoss Domain Controller Host Name
- **omc_management_port**: Your JBoss Management Console Port

For Non-Secure (no-SSL):

- CredType:MonitorCreds

Properties:

- **user_name**:Your JBoss Management User Name
- **password**:Your JBoss Management User Password
- **app_user_name**: Your JBoss Application User Name
- **app_user_password**: Your JBoss Application User Password
- CredType:AliasCredential
- **Alias**: Your JBoss Management User Name
- **Password**: Your JBoss Application User Password

For Secure (SSL):

- CredType:MonitorCreds

Properties:

- **user_name**: Your JBoss Management User Name
- **password**: Your JBoss Application User Password
- **app_user_name**: Your JBoss Application User Name
- **app_user_password**: Your JBoss Application User Password
- **ssl_trust_store**: Your cloud agent Truststore Location
- **ssl_trust_store_password**: Your cloud agent Truststore password.
- CredType:AliasCredential
- **Alias**: Your JBoss Management User Name
- **Password**: Your JBoss Application User Password
- **CredType**: Store Credential
- **StoreLocation**: Your OMC Cloud Agent Truststore Location

JBoss Server/Domain JSON Files and Properties

- **StorePassword:** Your OMC Cloud Agent Truststore Password
-

JBoss Server/Domain UI Fields

JBoss Server

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Your Fully-qualified JBoss Standalone J2EE Server Host Name
- **JBoss Management Port:** Your JBoss Management Console Port
- **Cloud Agent:** Cloud agent monitoring the JBoss Server/Domain.

Monitoring Credentials (JBoss Credentials)

- **Username:** Your JBoss Management User Name
 - **Password:** Password used for authentication.
-

Oracle Coherence

Oracle Coherence JSON Files and Properties

Definition File: **omc_oracle_coherence.json**

- **omc_jmx_port** - Coherence JMX port
- **omc_machine_name** - Coherence management node host.
- **omc_skip_cache_discovery** - Specify that Coherence

Credential Files

omc_oracle_coherence_cred.json

coherence_credentials.json

If Coherence is configured using a secured JMX connection, then a credentials file has to be passed as an input argument.

- **omc_username** - JMX connection username.
 - **omc_password** - JMX connection password.
-

Oracle Coherence UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **JMX Remote Port:** Coherence JMX port
- **JMX Management Server Machine Name:** Coherence management node host.
- Do not discover caches: If set to True, new Coherence cache targets will not be discovered. This is recommended for clusters with a very large number of caches (over 1000).
- **Cloud Agent:** Cloud agent monitoring the host where Oracle Coherence is installed.

Monitoring Credentials (Coherence Credentials)

- **Username:** JMX connection username.
 - **Password:** JMX connection password.
-

Microsoft Internet Information Services

Microsoft Internet Information Services JSON Files and Properties

Definition Files

omc_microsoft_iis_server_local_sample.json

- **host_name:** Hostname of Microsoft IIS Server
- **install_dir:** Absolute installation path of the Microsoft IIS Server. You need to specify the path using double backslashes (\\).
Example: C:\\Windows\\system32\\inetsrv

omc_microsoft_iis_server_remote_sample.json

- **omc_is_remote:** Property to indicate if the Microsoft IIS Server is local(no) or remote(yes)

Credential File: **omc_microsoft_iis_server_remote_creds_sample.json**

Credential properties (Applicable for remote monitoring via WMI)

- **wbem_username:** Windows user on the Microsoft IIS Server host
 - **wbem_password:** Password for the Windows user
-

Microsoft Internet Information Services UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Hostname of Microsoft IIS Server
- **Installation Directory:** Absolute installation path of the Microsoft IIS Server. You need to specify the path using double backslashes (\\).
Example: C:\\Windows\\system32\\inetsrv
- **Logging Directory:** Absolute path to log file directory.
- **Cloud Agent:** Cloud agent monitoring the host where Microsoft Internet Information Services is installed.

Monitoring Credentials

- **Host Username:** Windows user on the Microsoft IIS Server host
 - **Host Password:** Password for the Windows user
-

Oracle Unified Directory

Oracle Unified Directory JSON Files and Properties

- i) OUD Server
- ii) OUD Proxy Server

Definition Files

omc_oud_directory.json

omc_oud_proxy.json

Credential File: **omc_oud_creds.json**

Replace any text inside brackets <> excluding these brackets with your values according the legend inside <>

- **Administration Port:** The administration port of the target server instance.
- **Directory Server Host:** The fully qualified domain name of the target server instance. For replicated servers, you must provide the same host name that was used when replication was configured.
- **Trust All :** Set to true by default. This implies that all the certificates that are presented by the server (or servers, in the case of replication) will be accepted automatically. Change this setting if you want to specify different behavior. (Optional)
If you have changed the default setting for the Trust All field, enter a path in the Trust Store Path field..

The agent will use the trust store located in this path to validate the certificates of the administration connector that are presented by the server(s). This path must be readable by the agent (and thus located in a file system that is accessible by the agent). The trust store must contain the public keys of the administration connector certificates. It must be in JKS format and must not be password protected.

Oracle Internet Directory (OID)/Oracle Access Manager (OAM)

Oracle Internet Directory (OID)/Oracle Access Manager (OAM) JSON Files and Properties

Definition File: **omc_weblogic_domian.json**

- **displayName:** WebLogic Domain Entity Display Name that is displayed in the Infrastructure Monitoring UI time zone.
- **Region:** Time Zone (tz database time zones). For example: America/New_York.
- **port:** Port used for the WebLogic Admin Server(Console)
- **protocol:** The Protocol used for the WebLogic Server. For example: t3
- **admin_server_host:** Fully qualified WebLogic Admin Server Host Name where the WebLogic Admin Server is installed.

Credential File: **omc_weblogic_domain_creds.json**

- **user_name:** WebLogic Domain Entity User Name.
 - **password:** WebLogic Domain Entity Password.
-

Oracle Identity Manager (OIM)

Oracle Identity Manager (OIM) JSON Files and Properties

Definition File: **omc_weblogic_domian.json**

- **displayName:** WebLogic Domain Entity Display Name that is displayed in the Infrastructure Monitoring UI time zone.
- **Region:** Time Zone (tz database time zones). For example: America/New_York.
- **port:** Port used for the WebLogic Admin Server(Console)
- **protocol:** Protocol used for the WebLogic Server. For example: t3
- **admin_server_host:** Fully-qualified WebLogic Admin Server Host Name where the WebLogic Admin Server is installed.

Credential File: **omc_weblogic_domain_creds.json**

- **user_name:** WebLogic Domain Entity User Name.
 - **password:** WebLogic Domain Entity Password.
-

Oracle Identity Manager (OIM) UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Port:** Port used for the WebLogic Admin Server(Console)
- **Protocol:** Protocol used for the WebLogic Server. For example: t3
- **Administration Server Host:** Fully-qualified WebLogic Admin Server Host Name where the WebLogic Admin Server is installed.
- **Discover Coherence:** Discover Oracle Coherence.
- **Cloud Agent:** Cloud agent monitoring the host where OIM is installed.

Monitoring Credentials (WebLogic Server Credentials)

- **Username:** WebLogic Domain Entity User Name.
 - **Password:** WebLogic Domain Entity Password.
-

Oracle Clusterware (CRS)

Oracle Clusterware (CRS) JSON Files and Properties

Definition File: **omc_oracle_clusterware_sample.json**

- **scan_name :** Scan name for the cluster
- **cluster_name :** Cluster name
- **scan_port :** Scan port for the cluster
- **oracle_home :** CRS home base directory
- **omc_sshd_port :** SSH port value for remote monitoring
- **credential_ref :** "credentialRefs":["remote_sshcreds"] → for SSH Key based authentication

Credential Files

omc_oracle_clusterware_credless_sample.json

omc_oracle_clusterware_credential_sample.json

- **SSHUserName:** Your SSH user used to remotely log onto the listener host
 - **SSHUserPassword :** Your SSH host Password. Optional , if there is a passwordless SSH setup. In this case, provide a private key field
 - **SSH_PVT_KEY:** Path of your private key file. This private key is optional if the keys are generated at default location <user home>/.ssh
 - **sshdHost:** Your Cluster Host Name
 - **sshdPort:** SSH port
-

Oracle E-Business Suite (EBS)

Oracle E-Business Suite (EBS) JSON Files and Properties

Definition File: **omc_oracle_ebiz_sample.json**

- **name:** Oracle EBS Entity Name
- **displayName:** This is Oracle EBS Entity Display Name shown in the Infrastructure Monitoring UI.
- **timezoneRegion:** Time Zone Example: PDT, GMT, etc
- **omc_ebs_db_host:** Fully-qualified Host Name where the Oracle Database is installed.
- **omc_ebs_db_port:** Oracle Database port
- **omc_ebs_db_service_name:** Oracle Database Service Name

Credential File: **omc_oracle_ebiz_sample_creds.json**

- **DBUserName:** Oracle Database username
 - **DBPassword:** Oracle database user's password
 - **user_name:** WebLogic Domain Entity User Name
 - **password:** WebLogic Domain Entity Password
-

Oracle E-Business Suite (EBS) UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Database Host Name:** Host name of the database containing the EBS schema.
- **Database Port:** Port used to connect to the database containing the EBS schema.
- **Database Service Name:** Service name of the database containing the EBS schema.
- **Cloud Agent:** Cloud agent monitoring the host where EBS is installed.

Monitoring Credentials

Database Credentials

- **Username:** Database user who has the necessary privileges on the underlying V\$views such as moncs, or the monitoring user.
- **Password:** Password used for authentication.
- **Database Role**

WebLogic Server Credentials

- **Username:** WebLogic Server user with at least the Monitor security role.
 - **Password:** WebLogic Domain Entity Password
-

Generic Metric Collector

Generic Metric Collector JSON Files and Properties

Definition File: **omc_generic_metric_collector_collectd_auto_map_sample.json**

- **name:** Your name for the collectd collector. Eg. collectd-host1.example.com
- **displayName:** Your display name for the collectd collector Eg. collectd-host1.example.com

Property Value Inputs:

- **host_name:** Name of the host where collectd is installed. For example, host1.example.com
- **omc_query_interface_path:** Location where collectdctl is installed. For example: "/opt/collectd/bin/collectdctl"
- **omc_filter_expression:** "\${\$.[?(@.host=='<Value of the host field in the metric payload sent by collectd>')]}" For example: "\${\$.[?(@.host=='host1.example.com')]}"

Definition File: **omc_generic_metric_collector_collectd_manual_map_sample.json**

Additional Property Value Inputs(manual map case)

- **omc_mapping_metadata_file_path:** Path to the mapping metadata json file. For example: /scratch/agent/gmc/mapping_metadata_processes.json
-

Oracle GoldenGate

Oracle GoldenGate JSON Files and Properties

 **Note:**

Credentials are required for both Oracle GoldenGate Microservice and Oracle GoldenGate OCI architectures. No credentials are required for Oracle GoldenGate Classic architecture.

Definition Files

omc_oracle_goldengate_sample_arch_classic.json

omc_oracle_goldengate_sample_arch_microservice.json

omc_oracle_goldengate_sample_arch_oci.json

- **host_name:** Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **omc_port:** Oracle GoldenGate Service Manager port (if Microservice architecture). Performance Metric port if available, else Manager Port (if Classic architecture). Port to connect to OCI GoldenGate Service instance (if OCI GoldenGate architecture).
- **omc_ogg_arch:** Architecture - Microservice, Classic or OCI
- **omc_ogg_conn_timeout:** Connection Timeout in Seconds (Default 15 sec)

Credential File: **omc_oracle_goldengate_sample_creds.json**

Credentials (Microservice and OCI architecture only)

- **Alias:** Oracle GoldenGate Username
 - **Password:** Oracle GoldenGate Password
-

Oracle GoldenGate UI Fields

 **Note:**

Credentials are required only for Oracle GoldenGate Microservice architecture. No credentials are required for Oracle GoldenGate Classic architecture.

Classic

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **Port:** Oracle GoldenGate Service Manager port (if Microservice architecture). Performance Metric port if available, else Manager Port (if Classic architecture). Port to connect to OCI GoldenGate Service instance (if OCI GoldenGate architecture)
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where Oracle GoldenGate is installed.

Microservice

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **Service Manager Port:** Oracle GoldenGate Service Manager port (if Microservice architecture). Otherwise, Performance Metric port if available, else Manager Port (if Classic architecture)
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where Oracle GoldenGate is installed.

OCI GoldenGate

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **Port:** Port to connect to OCI GoldenGate Service instance, for example, 443.
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where Oracle GoldenGate is installed. The agent needs to be version 1.60 or higher.

Monitoring Credentials (Oracle GoldenGate Credentials)

- **Username:** Oracle GoldenGate Username
 - **Password:** Oracle GoldenGate Password
-

C

Discovery

This appendix contains how-to instructions for discovering various entity types to Oracle Infrastructure Monitoring.

Topics:

- [Add Apache HTTP Server](#)
- [Add Apache SOLR](#)
- [Add Apache Zookeeper](#)
- [Add Arbor CP](#)
- [Add Arbor TMS](#)
- [Add Arista Ethernet Switch](#)
- [Add Brocade Fibre Channel Switch](#)
- [Add Apache Cassandra Database](#)
- [Add Cisco Catalyst Switch](#)
- [Add Cisco Nexus Ethernet Switch](#)
- [Add Docker Engine/Docker Container](#)
- [Add Docker Swarm](#)
- [Add F5 BIG-IP DNS](#)
- [Add F5 BIG-IP LTM](#)
- [Add Hadoop Cluster](#)
- [Add JBoss Server/Domain](#)
- [Add Juniper Ethernet Switch](#)
- [Add Juniper MX Router](#)
- [Add Juniper Netscreen Firewall](#)
- [Add Juniper SRX Firewall](#)
- [Add Kubernetes Cluster](#)
- [Add Microsoft IIS](#)
- [Add Microsoft SCOM](#)
- [Add Microsoft SQL Server](#)
- [Add MongoDB](#)
- [Add MySQL Database](#)
- [Add NetApp FAS](#)
- [Add NGINX](#)
- [Add Oracle Access Manager/Oracle Internet Directory](#)

- Add Oracle Automatic Storage Management (ASM)
- Add Oracle Clusterware (CRS)
- Add Oracle Coherence Clusters
- Add Oracle Database Listener Cluster
- Add Oracle Database Listeners
- Add Oracle Databases
- Add Oracle ES2 Ethernet Switches
- Add Oracle GoldenGate
- Add Oracle HTTP Server
- Add Oracle Identity Manager
- Add Oracle Infiniband Switch
- Add Oracle JVM Runtime
- Add Oracle NoSQL Database
- Add Oracle Pluggable Database
- Add Oracle Power Distribution Unit (PDU)
- Add Oracle Service Bus
- Add Oracle Traffic Director
- Add Oracle Unified Directory
- Add Oracle Virtual Networking
- Add Oracle VM Manager
- Add Oracle VM Server for SPARC (LDOMS)
- Add Oracle WebLogic Server/Domain
- Add SPARC/Intel Computers
- Add Tomcat
- Add VMware vCenter
- Add ZFS Storage Appliance

Add Apache HTTP Server

You can add Apache HTTP Server entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Apache HTTP Server for monitoring.

Apache HTTP Server Prerequisites

In this release, only Apache HTTP Server 2.4.x and 2.2 for Linux are supported.

To enable the collection of configuration metrics, note the following:

1. The Cloud Agent should be installed on the same host as Apache HTTP Server. The Apache *.conf file(s), including `httpd.conf` file, should be accessible and readable by the Cloud Agent install user.
2. The Apache install user and the Cloud Agent install user should be a part of the same operating system group.

In order to monitor an Apache HTTP Server you must first:

- Enable 'mod_status' for the Apache module.
- Configure/server-status location directive for the specified Host and Port (default or configured virtual host).
- Turn 'ON' the Extended Status.
- If applicable, provide access to the configured location directive so that HTTP/HTTPS request can be successfully made from the host where the agent is installed on.

For more information, see https://httpd.apache.org/docs/2.4/mod/mod_status.html and <http://httpd.apache.org/docs/current/mod/core.html#location>.

For HTTPS/Secure communication between Apache HTTP Server and the cloud agent during metrics collection, you must provide an SSL certificate. To make the certificate available with the cloud agent:

1. Append the contents of your certificate file to the existing certificate file. For example, on a UNIX host the existing certificate file is: `<AGENT_BASE_DIR>/sysman/config/b64InternetCertificate.txt`

Ensure that only the following lines are appended to the `b64InternetCertificate.txt` file. Do not include blank lines, comments, or any other special characters.

```
----BEGIN CERTIFICATE----  
<<<Certificate in Base64 format>>>  
----END CERTIFICATE----
```

2. Restart the agent by running the following commands from the agent installation directory (for example, on a UNIX host, this directory is `<AGENT_BASE_DIR>/agent_inst/bin`).

- a) `./omcli stop agent`
- b) `./omcli start agent`

For data retrieval of memory-related metrics (supported on Unix platforms and when an entity is locally monitored), the PID file (`httpd.pid`) file needs to be accessed.

If Apache is running as `root` or some user other than the agent process owner, access to the PID file will fail. Hence, to allow access to `httpd.pid`, you need to ensure that the file can be accessed without compromising Linux security. There are several ways to achieve this. One option is as follows:

Apache HTTP Server Prerequisites

As a privileged user, run the following commands:

```
setfacl -R -L -d -m u:<agent_user>:rx /etc/httpd/run
setfacl -R -L -m u:<agent_user>:rx /etc/httpd/run
```

where `/etc/httpd/run` is the directory containing the PID file.

Step 2: Decide how you want to add the Apache HTTP Server.

You can add Apache HTTP Server entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Apache HTTP Server Entity Type.
3. Enter the following UI properties.

Apache HTTP Server UI Properties

- **Entity Name:** Name of the Apache HTTP Server entity appearing in the UI. .
 - **Host Name:** Host Name of the Apache HTTP Server.
 - **Server Root:** Server Root of the Apache HTTP Server.
 - **Absolute Path of `httpd.conf`:** Absolute path of the Apache `httpd.conf` file. Note: Filename needs to be appended.
 - **Is Remote:** Is the Apache installation host different from the agent host? Yes/No
 - **Binary Home:** Absolute path of the `httpd` binary (Optional - default value, if not provided: `<Apache Home>/bin` or `/usr/bin`)
 - **Protocol:** Protocol for connection to the Apache HTTP Server.
 - **Listen Port:** Listen Port of the Apache HTTP Server.
 - **Server Status Connection Hostname:** Server-status (Optional). This property specifies the value for the Host Name configured for `/server-status` connection (if different than FQDN - e.g., `localhost`). This needs to be specified if the connection-string (`host:port`) has a different host-name value than the value specified for `host_name` property. If specified, this value will be used to connect to Apache and retrieve data from the URI `/server-status`. If this property value is not specified, the default value will be the same as the `host_name` property value. (Optional)
 - **Cloud Agent:** Cloud agent monitoring the Apache HTTP Server.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Apache HTTP Server JSON Files and Properties

Definition File: **omc_generic_apache_sample.json**

- **host_name**: Host Name of the Apache HTTP Server.
 - **omc_listen_port**: Listen Port of the Apache HTTP Server.
 - **omc_httpd_conf_path**: Absolute Path of httpd.conf
 - **omc_protocol**: Protocol for connection to the Apache HTTP Server.
 - **omc_server_root**: Server Root of the Apache HTTP Server.
 - **omc_is_remote**: Indicates whether the HTTP Apache Server is local(NO) or remote(YES) - possible values: yes / no.
 - **omc_binary_home**: Absolute path of the httpd binary (Optional - default value, if not provided: \$omc_server_root/bin)
 - **omc_access_log_path**: Access Log Path (Optional)
 - **omc_error_log_path**: Error Log Path (Optional)
 - **omc_server_status_connect_host**: Server-status (Optional). This property specifies the value for the Host Name configured for /server-status connection (if different than FQDN - e.g., localhost). This needs to be specified if the connection-string (host:port) has a different host-name value than the value specified for host_name property. If specified, this value will be used to connect to Apache and retrieve data from the URI /server-status. If this property value is not specified, the default value will be the same as the host_name property value. (Optional)
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Apache HTTP Server, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Apache SOLR

You can add Apache SOLR entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Apache SOLR for monitoring.

Prerequisites

Two modes are supported: standalone & solrcloud

Monitoring is done over REST APIs exposed by Apache SOLR

Monitoring credentials require read access to following URIs:

- `/admin/collections?action=clusterstatus`
- `/admin/collections?action=overseerstatus`
- `/admin/info/system`
- `/admin/info/threads`
- `/admin/cores`
- `/<core_name>/admin/mbeans`

Credentials

1. Without Credentials:
 - a. non-secure (http)
 - b. secure (https)
2. With Credentials:
 - a. Client Authentication - (2-way SSL)
 - b. Basic Authentication - non secure
 - c. Basic Authentication - secure
 - d. Basic Authentication with Client authentication

Step 2: Decide how you want to add the Apache SOLR entity.

You can add Apache SOLR entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Apache SOLR Entity Type.
3. Enter the following UI properties.

Apache SOLR UI Fields

- **Discover Using Credentials:** Discover Apache SOLR using Apache SOLR credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Base URL:** The base URL for REST API invocations.
- **Cloud Agent:** Cloud agent monitoring the host where the Apache SOLR is installed.

Monitoring Credentials

Basic Authentication

- **Username:** Apache SOLR username.
- **Password:** Apache SOLR password.

Client Authentication

- **Keystore Location:** Full path to the location of the keystore file.
- **Keystore Password:** Keystore authentication password.

BasicClient

Basic Authentication

- **Username:** Apache SOLR username.
- **Password:** Apache SOLR password.

Client Authentication

- **Keystore Location:** full path to the location of the keystore file.
 - **Keystore Password:** Keystore authentication password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Apache SOLR JSON Files and Properties

Definition Files:

omc_solr_instance_credless.json
omc_solr_instance_creds.json
omc_solrcloud_credless.json
omc_solrcloud_creds.json

Credential Files:

solr_basic_authentication.json
solr_client_authentication.json
solr_client_with_basic_authentication.json

Replace any text inside brackets <> excluding these brackets with your values according to the legend inside <>

Notes:

- a. Credential JSONs are same for both standalone (omc_solr_instance) & solrcloud (omc_solrcloud)
- b. For secure solr standalone & solrcloud (with or without creds), server certificates for all the instances need to be added to the agent keystore using following command
omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc <certificate location> -alias <alias>
- c. For client authentication the type of the agent keystore is JKS keystore

Examples of Base URLs:

non secure – http://<hostname>:<port>/solr/
secure — https://<hostname>:<port>/solr/

Example of credentialRefs in the entity JSONs:

```
"credentialRefs" : ["SolrKeyStoreCredRef"] ---- for client authentication
"credentialRefs" : ["SolrBasicCredRef"] ---- for basic authentication
"credentialRefs" : ["SolrBasicCredRef", "SolrKeyStoreCredRef"] ---- for
client with basic authentication
```

3. Add the entity using omcli.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Apache SOLR, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Apache Zookeeper

You can add Apache Zookeeper entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Decide whether you want to add Apache Kafka.

Apache Kafka discovery is integrated with Apache Zookeeper discovery and cannot be added by itself. If you want to include Apache Kafka, you need to perform prerequisite configuration tasks. See [Apache Kafka](#) for details.

Note:

If the Apache Kafka prerequisite configuration tasks are not performed, Zookeeper discovery will proceed and will not be impacted. Details on the missing Kafka prerequisites can be viewed by running the `omcli status_entity -verbose` option with WARNING level severity.

Step 2: Decide how you want to add the Apache Zookeeper.

You can add Apache Zookeeper entities using one of the following methods:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Apache Zookeeper Entity Type.
3. Enter the following UI properties.

Apache Zookeeper UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
 - **Host Name:** Zookeeper host name.
 - **Zookeeper listening port:** Zookeeper port used to listen for client connections.
 - **Cloud Agent:** Cloud agent monitoring the host where Apache Zookeeper is installed.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Apache Zookeeper JSON Files and Properties

Definition File: **omc_apache_zookeeper.json**

- **listen_port**: Zookeeper port used to listen for client connections.
 - **host_name**: Zookeeper host name.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Apache Zookeeper, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Arbor CP

You can add Arbor CP entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Arbor CP for monitoring.

Prerequisites

SNMP v1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during Arbor appliance configuration along with IP address of Cloud Agent which will be used for appliance monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the appliance..

Read-only access is adequate for Arbor appliance monitoring.

Step 2: Decide how you want to add Arbor CP.

You can add Arbor CP entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Arbor CP Entity Type.
3. Enter the following UI properties.

Arbor CP Firewall UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** `snmp://<Fully qualified host name or IP address of the Arbor CP Firewall>`
- **SNMP Port:** Port where the Arbor CP Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Arbor CP Firewall is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Arbor CP Firewall JSON Files and Properties

Definition File: **omc_arbor_cp_sample.json**

- **omc_dispatch_url**: snmp://<Fully qualified host name or IP address of the Arbor CP Firewall>
- **omc_snmp_port**: Port where the Arbor CP Firewall listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout**: Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version**: SNMP version used to monitor the Arbor CP Firewall (2c or 3) - 2c by default (optional)

Credential Files

omc_arbor_cp_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY**: SNMPv2c community string

omc_arbor_cp_snmpv3_sample_creds.json

SNMP v3

- **authUser**: SNMPv3 username.
 - **authPwd**: Password used for authentication.
 - **authProtocol**: Protocol used for authentication (MD5 or SHA).
 - **privPwd**: Password used for encryption.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Arbor CP, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Arbor TMS

You can add Arbor TMS entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Arbor TMS for monitoring.

Prerequisites

SNMP v1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during Arbor appliance configuration along with IP address of Cloud Agent which will be used for appliance monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the appliance..

Read-only access is adequate for Arbor appliance monitoring.

Step 2: Decide how you want to add Arbor TMS.

You can add Arbor TMS entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Arbor TMS Entity Type.
3. Enter the following UI properties.

Arbor TMS Firewall UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** `snmp://<Fully qualified host name or IP address of the Arbor TMS Firewall>`
- **SNMP Port:** Port where the Arbor TMS Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Arbor TMS Firewall is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Arbor TMS Firewall JSON Files and Properties

Definition File: **omc_arbor_tms_sample.json**

- **omc_dispatch_url**: snmp://<Fully qualified host name or IP address of the Arbor TMS Firewall>
- **omc_snmp_port**: Port where the Arbor TMS Firewall listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version**: SNMP version used to monitor the Arbor TMS Firewall (2c or 3) - 2c by default (optional)

Credential Files

omc_arbor_tms_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY**: SNMPv2c community string

omc_arbor_tms_snmpv3_sample_creds.json

SNMP v3

- **authUser**: SNMPv3 username.
 - **authPwd**: Password used for authentication.
 - **authProtocol**: Protocol used for authentication (MD5 or SHA).
 - **privPwd**: Password used for encryption.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Arbor TMS, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Arista Ethernet Switch

You can add Arista Ethernet Switch entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Arista Ethernet Switch for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string (which was entered during the Arista Switch configuration) along with IP address of agent that will be used for Arista Switch monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus the authentication method (SHA or MD5) and authorization password if authorization is used. In addition, you must supply the privilege method (only DES is supported) and privilege password if privilege is used. Everything needs to be manually configured up front in the Arista Switch.

Read-only access is all that's required for Arista Switch monitoring.

Step 2: Decide how you want to add the Arista Ethernet Switch.

You can add Arista Ethernet Switch entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Arista Ethernet Switch Entity Type.
3. Enter the following UI properties.

Arista Ethernet Switch UI Fields

- **Entity Name:** Name of your Arista Ethernet Switch in Oracle Management Cloud
- **Dispatch URL:** `snmp://<Fully qualified host name or IP address of Arista Ethernet Switch>`
- **SNMP Port:** Port where Arista Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Arista Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **COMMUNITY:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication - supply either MD5 or SHA
 - **Privacy Password:** Password used for encryption.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Arista Ethernet Switch JSON Files and Properties

Definition File: `omc_arista_eth_switch_sample.json`

- **omc_dispatch_url**: `snmp://<Fully qualified host name or IP address of Arista Ethernet Switch>`
- **omc_snmp_port**: Port where Arista Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version**: SNMP version used to monitor Arista Ethernet Switch (2c or 3) - 2c by default (optional)

Credential Files

Choose the credential json file according to what SNMP credentials you'd like to use - SNMP v2c or SNMP v3. SNMP v2c

omc_arista_eth_switch_snmpv2_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2. Under "value", within the square brackets, provide the SNMPv2c community string used during the Arista Ethernet Switch configuration.

- **COMMUNITY**: SNMPv2c community string.

omc_arista_eth_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3

- **authUser**: Under "value", within the square brackets, provide provide SNMPv3 user name.
- **authPwd**: Under "value", within the square brackets, provide the auth password or empty out the field.
- **authProtocol**: Under "value", within the square brackets, provide the auth-method (SHA or MD5).
- **privPwd**: Under "value", within the square brackets, provide the privilege method password, if privilege is used. Only the DES privilege method is supported.

-
3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Arista Ethernet Switch, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Brocade Fibre Channel Switch

You can add Brocade Fibre Channel Switch entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Brocade Fibre Channel Switch for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string (entered during Brocade Fibre Channel Switch configuration), along with the IP address of the agent that will be used for Brocade Fibre Channel Switch monitoring.

If SNMPv3 is used, you must provide the SNMPv3 the user, plus the authentication method (SHA or MD5) and authorization password (if authorization is used), plus privilege method (only DES is supported) and privilege password if a privilege method is used. All of this needs to be manually configured up front in the Brocade Fibre Channel Switch.

Read-only access is enough for Brocade Fibre Channel Switch monitoring.

Step 2: Decide how you want to add the Brocade Fibre Channel Switch.

You can add Brocade Fibre Channel Switch entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Brocade Fibre Channel Entity Type.
3. Enter the following UI properties.

Brocade Fiber Channel Switch UI Fields

- **Entity Name:** Name of your Brocade Fiber Channel Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Brocade Fiber Channel Switch>
- **SNMP Port:** Port where Brocade Fiber Channel Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring your Brocade Fiber Channel Switch.

Monitoring Credentials

SNMP V1/V2:

- **Community String:** SNMPv1/v2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Brocade Fiber Channel Switch JSON Files and Properties

Definition File: **omc_brocade_fc_switch_sample.json**

- **name:** Your Brocade Fibre Channel Switch entity name.
- **displayName:** Your Brocade Fibre Channel Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under “value”, provide fully qualified host name or IP address of the Brocade Fibre Channel Switch
- **omc_dispatch_url:** Under “value”, following the string snmp://, provide the fully qualified hostname or IP address of the Brocade Fibre Channel Switch.
- **omc_snmp_port:** Under “value”, provide the port where the Brocade Fibre Channel Switch listens for SNMP requests. The default is 161.
- **omc_snmp_timeout:** Under “value”, provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under “value”, provide the SNMP version used to monitor the Brocade Fibre Channel Switch.

*Credential Files***omc_brocade_fc_switch_snmpv1_sample_creds.json**

Use this credential file if you have configured your switch with SNMPv1/v2.

- **community:** Under “value”, within the square brackets, provide the SNMPv2c community string used during the Brocade Fibre Channel Switch configuration.

omc_brocade_fc_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3.

- **authUser:** Under “value”, within the square brackets, provide the SNMPv3 username.
 - **authPwd:** Under “value”, within the square brackets, provide the authorization password or empty out the field. .
 - **authProtocol:** Under “value”, within the square brackets, provide the authorization method (SHA or MD5).
 - **privPwd:** Under “value”, within the square brackets, provide the privilege method password, if privilege is used. Only the DES privilege method is supported.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Brocade Fibre Channel Switch, see the following:

- Lack of Data
- Create an Agent Support Bundle

Add Apache Cassandra Database

You can add Apache Cassandra Database entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Apache Cassandra Database for monitoring.

Prerequisites

The default settings for Cassandra make JMX accessible only from the local host. If you want to enable remote JMX connections, change the `LOCAL_JMX` setting in `cassandra-env.sh` and enable authentication and/or SSL. To do this, perform the following procedure:

1. Open the `cassandra-env.sh` file for editing and update or add these lines:

```
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.authenticate=true"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.password.file=/etc/
cassandra/jmxremote.password"
```

If the `LOCAL_JMX` setting is in your file, set it to **no**:

```
LOCAL_JMX=no
```

2. Depending on whether the JDK or JRE is installed:

- Copy the `jmxremote.password.template` from `/jdk_install_location/jre/lib/management/` to `/etc/cassandra/` and rename it to `jmxremote.password`

```
$ cp /jdk_install_dir/lib/management/jmxremote.password.template /etc/
cassandra/jmxremote.password
```
- Copy the `jmxremote.password.template` from `/jre_install_location/lib/management/` to `/etc/cassandra/` and rename it to `jmxremote.password`

```
$ cp /jre_install_dir/lib/management/jmxremote.password.template /etc/
cassandra/jmxremote.password
```

3. Change the ownership of the `jmxremote.password` to the user you use to run Cassandra and change permission to read-only:

```
$ chown cassandra:cassandra /etc/cassandra/jmxremote.password
$ chmod 400 /etc/cassandra/jmxremote.password
```

4. Edit `jmxremote.password` and add the user and password for JMX-compliant utilities:

```
monitorRole QED
controlRole R&D
cassandra cassandrapassword
```

 **Note:**

The Cassandra user and Cassandra password shown in the above sample are examples. Specify the user and password for your environment.

Prerequisites

5. Add the Cassandra user with read and write permission to `/jre_install_location/lib/management/jmxremote.access`

```
monitorRole readonly
cassandra readwrite
controlRole readwrite \
create javax.management.monitor.,javax.management.timer. \
unregister
```

6. Restart Cassandra.
-

Step 2: Decide how you want to add the Apache Cassandra Database.

You can add Apache Cassandra Database entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Apache Cassandra Database Entity Type.
3. Enter the following UI properties.

Apache Cassandra DB UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where Cassandra database is installed.
- **Port:** Apache Cassandra Database port.
- **Cloud Agent:** Cloud agent monitoring the host where the Apache Cassandra DB is installed.

Monitoring Credentials (Cassandra JMX Credentials)

- **Username:** Cassandra DB username.
 - **Password:** Password used for authentication.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Apache Cassandra DB JSON Files and Properties

Definition File: **omc_cassandra_db.json**

- **displayName**: This is Apache Cassandra Database Entity Display Name which is displayed in the Oracle Infrastructure Monitoring UI
- **timezoneRegion**: Time Zone Example: PDT, GMT
- **omc_url**: connection url to connect to the installed Apache Cassandra database; host:port
- **host_name** : Fully-qualified Host Name where Cassandra database is installed.
- **omc_port** : Apache Cassandra Database port.
- **cassandra_home**: Location of the Cassandra Installation directory.

Credential File: **omc_cassandra_db_creds.json**

- **DBUserName**: Cassandra DB username.
 - **DBPassword**: Cassandra DB user password.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Apache Cassandra Database, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Cisco Catalyst Switch

You can add Cisco Catalyst Switch entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Cisco Catalyst Switch for monitoring.

Prerequisites

To enable monitoring of the Cisco Ethernet (Catalyst) Switch, you will need to provide the SNMPv1/v2 or SNMPv3 credentials in the JSON credential file. Read-only access is sufficient for Cisco Catalyst Switch monitoring. For more information on how to configure an SNMP user for a Cisco Catalyst Switch, see http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960/software/release/12-2_55_se/configuration/guide/scg_2960/swsnmp.html#78160

Step 2: Decide how you want to add the Cisco Catalyst Switch.

You can add Cisco Catalyst Switch entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Cisco Catalyst Switch Entity Type.
3. Enter the following UI properties.

Cisco Catalyst Ethernet Switch UI Fields

- **Entity Name:** Name of your Cisco Catalyst Ethernet Switch in Oracle Management Cloud
- **Dispatch URL:** `snmp://<Fully qualified host name or IP address of Cisco Catalyst Ethernet Switch>`
- **SNMP Port:** Port where Cisco Catalyst Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Cisco Catalyst Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** The SNMPv2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication.
 - **Authorization Protocol:** Protocol used for authentication (MD5 or SHA)
 - **Privacy Password:** Password used for encryption.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Cisco Catalyst Ethernet Switch JSON Files and Properties

Definition File: **omc_cisco_eth_switch_sample.json**

- **name:** Your Cisco Ethernet (Catalyst) Switch entity name.
- **displayName:** Your Cisco Ethernet (Catalyst) Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **omc_dispatch_url:** Under "value", following the string snmp://, provide the fully qualified hostname or IP address of the Cisco Ethernet (Catalyst) Switch.
- **omc_snmp_port:** Under "value", provide the port where the Cisco Ethernet (Catalyst) Switch listens for SNMP requests, 161 by default.
- **omc_snmp_timeout:** Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under "value", provide the SNMP version used to monitor the Cisco Ethernet (Catalyst) Switch.

 **Note:**

This is an optional property which is used only with SNMPV1Creds and allowed values are "1" or "2c". The default value is "2c".

Do not remove the square brackets.

Credential Files

Choose the creds json file according to what SNMP credentials you'd like to use - SNMP v2c or SNMP v3.

omc_cisco_eth_switch_snmpv2c_sample_creds.json

- **COMMUNITY:** SNMPv2c community string

omc_cisco_eth_switch_snmpv3_sample_creds.json

- **authUser:** SNMPv3 username
- **authPwd:** password used for authentication
- **authProtocol:** protocol used for authentication - supply either MD5 or SHA
- **privPwd:** password used for encryption

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Cisco Catalyst Switch, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Cisco Nexus Ethernet Switch

You can add Cisco Nexus Ethernet Switch entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Cisco Nexus Ethernet Switch for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string (which was entered during Cisco Nexus Ethernet Switch configuration) along with IP address of agent that will be used for Cisco Nexus Ethernet Switch monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus the authentication method (SHA or MD5) and authentication password if authentication is used,. In addition, the privilege method (only DES supported) and privilege password must be supplied if privilege is used. Everything needs to be manually configured up front in the Cisco Nexus Ethernet Switch.

Read only access is enough for the Cisco Nexus Ethernet Switch monitoring.

Step 2: Decide how you want to add the Cisco Nexus Ethernet Switch.

You can add Cisco Nexus Ethernet Switch entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Cisco Nexus Ethernet Switch Entity Type.
3. Enter the following UI properties.

Cisco Nexus Ethernet Switch UI Fields

- **Entity Name:** Name of your Cisco Nexus Ethernet Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Cisco Nexus Ethernet Switch>
- **SNMP Port:** Port where Cisco Nexus Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Cisco Nexus Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** SNMPv2c community string

SNMP V3

- **Username:** SNMPv3 username
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication - supply either MD5 or SHA
 - **Privacy Password:** Password used for encryption.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Cisco Nexus Ethernet Switch JSON Files and Properties

Definition File: **omc_cisco_nexus_eth_switch_sample.json**

- **name:** Your Cisco Nexus Ethernet Switch entity name.
- **displayName:** Your Cisco Nexus Ethernet Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **omc_dispatch_url:** Under "value", following the string snmp://, provide the fully qualified hostname or IP address of the Cisco Nexus Ethernet Switch.
- **omc_snmp_port:** Under "value", provide the port where the Cisco Nexus Ethernet Switch listens for SNMP requests, 161 by default.
- **omc_snmp_timeout:** Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under "value", provide the SNMP version used to monitor the Cisco Nexus Ethernet Switch.

 **Note:**

This is an optional property which is used only with SNMPV1Creds and allowed values are "1" or "2c". The default value is "2c".

Credential Files

omc_cisco_nexus_eth_switch_snmpv2_sample_creds.json

- **COMMUNITY:** Use this credential file if you have configured your switch with SNMPv1/v2.

omc_cisco_nexus_eth_switch_snmpv3_sample_creds.json

Under "value", within the square brackets, provide the SNMPv3 user name.

- **authPwd:** Under "value", within the square brackets, provide the auth password or empty out the field.
- **authProtocol:** Under "value", within the square brackets, provide the auth-method (SHA or MD5).
- **privPwd:** Under "value", within the square brackets, provide the priv method password, if priv is used. Only the DES priv method is supported.

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Cisco Nexus Ethernet Switch, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Docker Engine/Docker Container

You can add Docker Engine/Docker Container entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Docker Engine/Docker Container for monitoring.

Docker Engine/Docker Container Configuration

You can configure a Docker Engine for monitoring in three ways:

Non-Secure Mode:

This mode doesn't need any credentials information. When the Docker Engine is configured in the non-secure mode (`http`), you simply need the Base URL to connect to the Docker Engine.

For example, a Base URL could be: `http://www.example.com:4243/`. Note the `http`, and not `https` mode.

To check if your Docker Engine is configured in non-secure mode, view the `/etc/sysconfig/docker` file. The following entries identify the Non-Secure Mode configuration:

```
http - non secure other_args="-H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock"
set proxy export HTTP_PROXY=<your proxy host>:80
```

You will need to provide the Docker Engine Base URL in the entity definition JSON file.

Secure Mode:

To check if your Docker Engine is configured in Secure Mode, view the `/etc/sysconfig/docker` file. If configured for:

- for 1-way SSL you will typically see an entry of the format:

```
https - secure 1 way SSL other_args="-H tcp://0.0.0.0:4243 -
H unix:///var/run/docker.sock --tls --tlscert=/<certificate
directory>/server-cert.pem --tlskey=/<certificate directory>/
server-key.pem"
```

- for 2-way SSL you will typically see an entry of the format:

```
https - secure 2 way SSL other_args="-H tcp://0.0.0.0:4243 -
H unix:///var/run/docker.sock --tlsverify --tlscacert=/
<certificate directory>/ca.pem --tlscert=/<certificate directory>/
server-cert.pem --tlskey=/<certificate directory>/server-key.pem"
```

If your Docker Engine is configured in Secure Mode, then you configure the monitoring credentials based on the type of communication defined.

- For **Secure 1-way SSL** you need to add the truststore *certificate* (CA certificate) in the cloud agent default truststore (`<agent home>/sysman/config/montrust/AgentTrust.jks`) using this command:

```
keytool -import -alias docker01 -keystore <agent home>/sysman/
config/montrust/AgentTrust.jks -file <directory of your Docker
certificate>/<certificate_file_name>.cer
```

Docker Engine/Docker Container Configuration

Use the password `welcome`. Note the `<agent home>` is the directory where the Cloud Agent was installed. See *Managing Cloud Agents in Oracle® Cloud Deploying and Managing Oracle Management Cloud Agents*.

You will only need to provide the Docker Engine Base URL in the entity definition JSON file.

- For **Secure 2–way SSL** you need to add the truststore *certificate* (CA certificate) and the *keystore* information in the agent default truststore (`<agent home>/sysman/config/montrust/AgentTrust.jks`).

1. Add the truststore *certificate*:

```
keytool -import -alias docker01 -keystore <agent home>/sysman/  
config/montrust/AgentTrust.jks -file <directory of your Docker  
certificate>/<certificate_file_name>.cer
```

Use the password `welcome`. Note the **agent home** is the directory where the Cloud Agent was installed.

2. Add the *keystore* information:

```
keytool -import -alias docker01 -keystore <agent home>/sysman/  
config/montrust/AgentTrust.jks -file <directory of your Docker  
certificate>/<certificate_file_name>.cer
```

Use the password `welcome`.

To add a Secure 2–way SSL Docker Engine entity you will need to create an entity definition JSON file along with a credentials JSON file. The entity definition JSON file will include your Docker Engine Base URL while the credentials file will have details about the credentials store and credentials.

For more information about how to create Docker certificates, see <https://docs.docker.com/engine/security/https/>.

Cloud Agent Configuration

If the cloud agent communicates with Oracle Management Cloud through a proxy (OMC_PROXYHOST & OMC_PROXYPORT parameters were set on the cloud agent when it was installed), Docker Engine / Docker Container discovery will fail. You'll need to perform additional configuration steps depending on the following situations:

For a New Agent Installation

If the agent requires proxy to communicate with Oracle Management Cloud, then use the gateway and set the proxy parameters (OMC_PROXYHOST & OMC_PROXYPORT) during gateway installation, and then set up the cloud agent (without proxy parameters) to point to the gateway.

For an Existing Agent

If the existing cloud agent has been set up to use the proxy to communicate with Oracle Management Cloud, to discover Docker Engine / Docker Container, execute the following commands on the cloud agent before performing entity discovery.

```
omcli setproperty agent -allow_new -name _configureProxyPerClient -  
value true  
omcli stop agent  
omcli start agent
```

Step 2: Decide how you want to add the Docker Engine/Docker Container.

You can add Docker Engine/Docker Container entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Docker Engine/Docker Container Entity Type.
3. Enter the following UI properties.

Docker Engine/Docker Worker UI Fields

- **Discover Using Credentials:** Discover using Docker Engine credentials (on by default).
- **Entity Name:** Your Docker Engine/Container name.
- **Base URL:** The base URL for REST API invocations.
- **Host Name:** The fully-qualified host name where the Docker Engine/Container is installed.
- **Swarm ID:** Unique identifier of the Docker Swarm containing the Docker Engine/Container.
- **Cloud Agent:** Cloud agent monitoring the host where the Docker Engine/Container is running.

Monitoring Credentials (Docker Engine Credentials)

- **Store Location:** The full path to the location of the keystore file.
 - **Store Password:** The keystore password to access the jks file.
 - **Store Type:** Currently, only JKS is supported.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Docker Engine/Docker Container JSON Files and Properties

Definition Files

omc_docker_engine_sample.json (used without the omc_docker_engine_sample_creds.json)

omc_docker_engine_secure_sample.json (used with the omc_docker_engine_sample_creds.json)

- **name:** Your Docker Engine name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **BaseURI:** Under “value”, provide the base URL for REST API invocations.
- **host_name:** Under “value”, provide the fully-qualified host name where the Docker Engine is installed.

Credential File: **omc_docker_engine_sample_creds.json**

- **StoreLocation:** Under “value”, within the square brackets, provide the full path to the location of the keystore file. You must have configured this entity security in the Prerequisite Tasks step. For example, <agent_home>/sysman/config/montrust/AgentTrust.jks
Note that in this release only jks file types are supported.
- **StorePassword:** Under “value”, within the square brackets, provide the keystore password to access the jks file.
Note that in this release only jksfile types are supported.

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Docker Engine/Docker Container, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Docker Swarm

You can add Docker Swarm entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Docker Swarm for monitoring.

Prerequisites and Credentials

Cloud Agent Configuration

If the cloud agent communicates with Oracle Management Cloud through a proxy (OMC_PROXYHOST & OMC_PROXYPORT parameters were set on the cloud agent when it was installed), Docker Swarm discovery will fail. You'll need to perform additional configuration steps depending on the following situations:

For a New Agent Installation

If the agent requires proxy to communicate with Oracle Management Cloud, then use the gateway and set the proxy parameters (OMC_PROXYHOST & OMC_PROXYPORT) during gateway installation, and then set up the cloud agent (without proxy parameters) to point to the gateway.

For an Existing Agent

If the existing cloud agent has been set up to use the proxy to communicate with Oracle Management Cloud, to discover Docker Swarm, execute the following commands on the cloud agent before performing entity discovery.

```
omcli setproperty agent -allow_new -name _configureProxyPerClient -
value true
omcli stop agent
omcli start agent
```

Credentials

There are three methods you can use to authenticate and connect to the Docker Swarm via Rest APIs

- 1) Non-secure
 - 2) Secure (https): 1-way SSL mode
 - 3) Secure (https): 2-way SSL mode
-

Step 2: Decide how you want to add the Docker Swarm.

You can add Docker Swarm entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Docker Swarm Entity Type.
3. Enter the following UI properties.

Docker Swarm UI Fields

- **Discover Using Credentials:** Discover using Docker Swarm credentials (on by default).
 - **Entity Name:** Your Docker Swarm name.
 - **Base URL of Swarm Leader:** The base URL of the Swarm Leader for REST API invocations.
 - **Host Name:** The fully-qualified host name where the Docker Swarm is installed.
 - **Cloud Agent:** Cloud agent monitoring the host where the Docker Swarm is running.
- Monitoring Credentials (Docker Swarm Credentials)
- **Store Location:** The full path to the location of the keystore file.
 - **Store Password:** The keystore password to access the JKS file.
 - **Store Type:** Currently, only JKS is supported.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Docker Swarm and Worker JSON Files and Properties

Entity JSONs for Docker Swarm:

Adding Non Secure Docker Swarm Target

Definition File: **Add_Entity_Docker_Swarm_Non_Secure.json**

Adding 1WAY Docker Swarm Target

Definition File: **Add_Entity_Docker_Swarm_1way_SSL.json**

Adding 2WAY Docker Swarm Target

Definition File: **Add_Entity_Docker_Swarm_2way_SSL.json**Credential File: **Docker_Swarm_Secure_Credentials.json****Entity JSONs for Docker Worker Engines:**

Adding Non Secure Docker Worker Engine

Definition File: **Add_Entity_Worker_Docker_Engine_Non_Secure.json**

Adding 1WAY Docker Worker Engine

Definition File: **Add_Entity_Worker_Docker_Engine_1way_SSL.json**

Adding 2WAY Docker Worker Engine

Definition: **Add_Entity_Worker_Docker_Engine_2way_SSL.json**Credential File: **omc_docker_engine_sample_creds.json**

For properties that should be updated, replace any text inside brackets <> excluding these brackets with your values according the legend inside <>

Examples of Base URLs:

NON SECURE MODE - http://<hostname>:<port>/ (Rest API URL for Invocation)

SECURE MODE- https://<hostname>:<port>/ (Rest API URL for Invocation)

For Basic Authentication:

"credentialRefs":["DockerSwarmCredRef"]

 **Note:**

The same Docker Engine credential JSON is used for Worker Engines.

For secure mode, in addition to configuring the JSONS, you need to add the Docker truststore certificate(CA certificate) to the Cloud Agent default truststore (\$EMSTATE/sysman/config/montrust/AgentTrust.jks). To do so, run the following command:

```
omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc
<certificate location> -alias dockercertificate
```

Example:

In the following example, slce03.cer is the CA certificate.

```
omcli secure add_trust_cert_to_jks -password welcome -
trust_certs_loc /home/sandepai/slce03.cert -alias dockercertificate
```

To fetch the Docker Swarm ID, do a GET on LEADER_BASE_URL/swarm

For example, do a GET on http://myserver.mycompany.com:4243/swarm

3. Add the entity using omcli.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```


4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Docker Swarm, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add F5 BIG-IP DNS

You can add F5 BIG-IP DNS entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare F5 BIG-IP DNS for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during F5 BIG-IP DNS configuration along with IP address of the Cloud Agent which will be used for the DNS monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the DNS.

Read-only access is adequate for DNS monitoring.

Step 2: Decide how you want to add the F5 BIG-IP DNS.

You can add F5 BIG-IP DNS entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the F5 BIG-IP DNS Entity Type.
3. Enter the following UI properties.

F5 BIG-IP DNS UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of F5 BIG-IP DNS>
- **SNMP Port:** Port where F5 BIG-IP DNS listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where F5 BIG-IP DNS is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using omcli and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

F5 BIG-IP DNS JSON Files and Properties

Definition File: **omc_f5_bigip_dns_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of F5 BIG-IP DNS>
- **omc_snmp_port:** Port where F5 BIG-IP DNS listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version:** SNMP version used to monitor F5 BIG-IP DNS (2c or 3) - 2c by default (optional)

Credential Files

omc_f5_bigip_dns_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_f5_bigip_dns_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd:** Password used for encryption.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of F5 BIG-IP DNS, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add F5 BIG-IP LTM

You can add F5 BIG-IP LTM entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare F5 BIG-IP LTM for monitoring.

Credentials

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during F5 BIG-IP LTM configuration along with IP address of Cloud Agent which will be used for the LTM monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the LTM.

Read-only access is adequate for LTM monitoring.

Step 2: Decide how you want to add the F5 BIG-IP LTM.

You can add F5 BIG-IP LTM entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the F5 BIG-IP LTM Entity Type.
3. Enter the following UI properties.

F5 BIG-IP LTM UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of F5 BIG-IP LTM>
- **SNMP Port:** Port where F5 BIG-IP LTM listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where F5 BIG-IP LTM is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using omcli and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

F5 BIG-IP LTM JSON Files and Properties

Definition File: **omc_f5_bigip_ltm_sample.json**

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of F5 BIG-IP LTM>
- **omc_snmp_port:** Port where F5 BIG-IP LTM listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version:** SNMP version used to monitor F5 BIG-IP LTM (2c or 3) - 2c by default (optional)

*Credential Files***omc_f5_bigip_ltm_snmpv2_sample_creds.json**

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

omc_f5_bigip_ltm_snmpv3_sample_creds.json

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd:** Password used for encryption.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of F5 BIG-IP LTM, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Hadoop Cluster

You can add Hadoop Cluster entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Hadoop Cluster for monitoring.

Prerequisites

By default, Hadoop runs in non-secure mode in which no actual authentication is required.

By configuring Hadoop to run in secure mode, each user and service needs to be authenticated by Kerberos in order to use Hadoop services.

To perform Kerberos authentication, the Cloud Agent requires the following:

1. `krb5.conf` file. This file can be found at `/etc/krb5.conf`
2. Username and password

The Cloud Agent can use only one `krb5.conf` at a time. If a single Agent needs to perform Kerberos authentication with more than one domain, these details should be defined in a single `krb5.conf` file.

Step 2: Decide how you want to add the Hadoop Cluster.

You can add Hadoop Cluster entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Hadoop Clusters Entity Type.
3. Enter the following UI properties.

Hadoop UI Fields

- **Discover Using Credentials:** Discover Hadoop using Hadoop credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Metric URL for NameNode 1:** Name Node 1 URL with port..
- **Metric URL for NameNode 2:** Name Node 2 URL with port.
- **Metric URL for Resource Manager 1:** Resource Manager Node 1 URL with port.
- **Metric URL for Resource Manager 2:** ResourceManager Node 2 URL with port.
- **Cloud Agent:** Cloud agent monitoring the host where Hadoop is installed.

Monitoring Credentials

SSL Trust Store

- **Store Location:** Path of the Truststore file.
- **Store Password:** The keystore password to access the JKS file.
- **Store Type:** Currently, only JKS is supported.

Alias

- **Alias Name:** Your alias name.
 - **Password:** Your alias password.
 - **Path of krb5.conf file:** Full path to the krb5.conf file.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Hadoop JSON Files and Properties

 **Note:**

Hadoop uses primary and secondary nodes to maintain availability. If one node goes down, the secondary node is used to fetch information. For both nodes, the primary and secondary roles can be switched at any given time. For this reason, you specify two nodes and two Resource Manager URLs without specifying whether they are primary or secondary.

Definition File: **hadoop_credless.json**

Use this file if Hadoop was configured with no credentials.

- **Name:** Hadoop entity name.
- **displayName:** Hadoop entity display name.
- **omc_nn1_metric_url** (Name Node 1): Name Node 1 URL with port.
- **omc_nn2_metric_url** (Name Node 2): Name Node 2 URL with port.
- **omc_rm1_metric_url** (Resource Manager 1): Resource Manager Node 1 URL with port.
- **omc_rm2_metric_url** (Resource Manager 2): ResourceManager Node 2 URL with port.

Example URL

```
http://<HOSTNAME>:<PORT>/ (Rest API URL for
    Invocation)
```

Definition File: **hadoop_creds.json**

- **name:** Hadoop entity name.
- **displayName:** Hadoop entity display name.
- **credentialRefs:** Hadoop credential information (hadoopTrustStore and hadoopSPNEGOCredentials) defined in the hadoop_credentials_input.json file.
- **omc_nn1_metric_url:** (Name Node 1): Name Node 1 URL with port.
- **omc_nn2_metric_url:** (Name Node 2): Name Node 2 URL with port.
- **omc_rm1_metric_url:** (Resource Manager 1): Resource Manager Node 1 URL with port.
- **omc_rm2_metric_url:** (Resource Manager 2): ResourceManager Node 2 URL with port.

Example URL

```
http://<HOSTNAME>:<PORT>/ (Rest API URL for Invocation)
```

Credential File: **hadoop_credentials_input.json**

- **hadoopTrustStore** consists of the following user-defined properties:
 - *StoreLocation:* Path of Truststore file.
 - *StorePassword:* Truststore Password
- **hadoopSPNEGOCredentials** consists of the following user-defined properties:
 - *Alias:* Alias name.
 - *Password:* Alias password.
 - *KRB5Conf:* Path of krb5.conf file.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Hadoop Clusters, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add JBoss Server/Domain

You can add JBoss entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare JBoss for monitoring.

Prerequisites


Before discovering a JBOSS server or domain, you must first add the JBOSS client jar file to the Cloud agent as a plug-in. The JBOSS client jar file contains the required JMX protocols that allow the agent to collect JBOSS metrics.

The JBOSS client jar is distributed as part of the JBOSS installation. When you download the JBOSS zip file, the client jar file will be bundled with it.

Step	Action
Step 1: Locate the JBOSS client jar file.	From the JBOSS home directory, you will find the client jar file at the following location: <pre>> JBOSS_HOME/bin/client</pre> In this directory, you'll see the <i>jboss-client.jar</i> file. This is the file you need to copy over to the Cloud agent location.
Step 2: Copy the JBOSS client jar file to the Cloud agent installation.	Copy the <i>jboss-client.jar</i> file to a secure location that is accessible by the Cloud agent. Typically, this is located on the same host where the agent is installed.

Step	Action
Step 3: Add the <code>jboss-client.jar</code> to the Cloud agent installation as a plug-in.	<p data-bbox="878 262 1385 325">From the Cloud agent home directory, navigate to the agent state directory:</p> <pre data-bbox="878 325 1385 367"><agent_home>/sysman/config</pre> <p data-bbox="878 367 1385 493">Create a classpath file. This file tells the agent where to find the <code>jboss-client.jar</code>. The file naming convention is <code><plugin_id>.classpath.lst</code>.</p> <p data-bbox="878 493 1385 588">Example: If you're adding the GFM plug-in (plug-in ID is <code>oracle.em.sgm</code>), the file name would be <code>oracle.em.sgm.classpath.lst</code>.</p> <p data-bbox="878 588 1385 682">Edit the classpath file and add the absolute path to the <code>jboss-client.jar</code> file at the end of the file.</p> <pre data-bbox="878 682 1385 745">/scratch/securelocation/jboss-client.jar</pre> <p data-bbox="878 745 1385 898">Bounce the agent. Any modifications made to the classpath file will not take effect until the agent is restarted. Once the agent has been bounced, you are ready to discover the JBOSS entity (server or domain).</p>

Step	Action
Step 4: Discover the JBOSS server/domain.	<ol style="list-style-type: none">1. From the Oracle Management Cloud console, select AdministrationàDiscoveryàAdd Entity. The Add Entity page displays.2. From the Entity Type drop-down menu, choose either JBOSS Domain or JBOSS Server. The appropriate JBOSS parameters are displayed.3. Enter the appropriate parameters and monitoring credentials.4. Click Add Entity. <p>About JBOSS Monitoring Credentials</p> <p>Depending on whether you choose JBOSS Server or JBOSS Domain entity type, the required monitoring credentials will differ:</p> <p><i>JBOSS Server</i></p> <ul style="list-style-type: none">• JBOSS Username: User account used by the agent for monitoring.• JBOSS Password: Password for the above user account. <p><i>JBOSS Domain</i></p> <ul style="list-style-type: none">• JBOSS Credentials:<ul style="list-style-type: none">– JBOSS username and password: Credentials used by the agent for monitoring.– App User Name and Password: Credentials used to communicate with servers in the domain.• User Credential Set:<ul style="list-style-type: none">– Alias and Password: Same as the JBOSS username and password used for the JBOSS Credentials.

 **Note:**

This is needed because two different fetchlets are being used.

Step 2: Decide how you want to add the JBoss.

You can add JBoss entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.

2. Select the JBoss Entity Type.
3. Enter the following UI properties.

JBoss Server/Domain UI Fields

JBoss Server

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Your Fully-qualified JBoss Standalone J2EE Server Host Name
- **JBoss Management Port:** Your JBoss Management Console Port
- **Cloud Agent:** Cloud agent monitoring the JBoss Server/Domain.

Monitoring Credentials (JBoss Credentials)

- **Username:** Your JBoss Management User Name
 - **Password:** Password used for authentication.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

JBoss Server/Domain JSON Files and Properties

Definition Files

omc_jboss_standalone_j2eeserver_sample.json

omc_jboss_standalone_j2eeserver_secure_sample.json

Credential Files

omc_jboss_standalone_j2eeserver_sample_creds.json

omc_jboss_standalone_j2eeserver_secure_sample_creds.json

JBoss Standalone Server:

- **host_name:** Your Fully-qualified JBoss Standalone J2EE Server Host Name
- **omc_management_port:** Your JBoss Management Console Port

For Non-Secure (no-SSL):

- CredType:MonitorCreds

Properties:

- **user_name:** Your JBoss Management User Name
- **password:** Your JBoss Management User Password

For Secure (SSL):

- CredType:MonitorCreds

Properties:

- **user_name:** Your JBoss Management User Name
- **password:** Your JBoss Management User Password
- **ssl_trust_store:** Your OMC Cloud Agent Truststore Location
- **ssl_trust_store_password:** Your OMC Cloud Agent Truststore Password

JBoss Domain:

- **omc_host_name:** Your Fully-qualified JBoss Domain Controller Host Name
- **omc_management_port:** Your JBoss Management Console Port

For Non-Secure (no-SSL):

- CredType:MonitorCreds

Properties:

- **user_name:**Your JBoss Management User Name
- **password:**Your JBoss Management User Password
- **app_user_name:** Your JBoss Application User Name
- **app_user_password:** Your JBoss Application User Password
- CredType:AliasCredential
- **Alias:** Your JBoss Management User Name
- **Password:** Your JBoss Application User Password

For Secure (SSL):

- CredType:MonitorCreds

Properties:

- **user_name:** Your JBoss Management User Name
 - **password:** Your JBoss Application User Password
 - **app_user_name:** Your JBoss Application User Name
 - **app_user_password:** Your JBoss Application User Password
 - **ssl_trust_store:** Your cloud agent Truststore Location
 - **ssl_trust_store_password:** Your cloud agent Truststore password.
 - CredType:AliasCredential
 - **Alias:** Your JBoss Management User Name
 - **Password:** Your JBoss Application User Password
 - **CredType:** Store Credential
 - **StoreLocation:** Your OMC Cloud Agent Truststore Location
 - **StorePassword:** Your OMC Cloud Agent Truststore Password
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE  
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of JBoss, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Juniper Ethernet Switch

You can add Juniper Ethernet Switch entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Juniper Ethernet Switch for monitoring.

Prerequisites

To enable monitoring, HTTP and SNMPv1/v2c/v3 are needed.

If SNMPv1/v2 is used, you must provide SNMP community string that has been used earlier in Juniper Switch configuration along with IP address of agent which will be used for Juniper Switch monitoring.

If SNMPv3 is used, in addition to SNMPv3 user, you must provide the auth method (SHA or MD5) and auth-password if auth is used, and priv method (only DES supported) and priv-password if priv used. You must configure everything manually in Juniper Switch. Read only access is sufficient for Juniper Switch monitoring.

Step 2: Decide how you want to add the Juniper Ethernet Switch.

You can add Juniper Ethernet Switch entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Juniper Ethernet Switch Entity Type.

3. Enter the following UI properties.

Juniper Ethernet Switch UI Fields

- **Entity Name:** Name of your Juniper Ethernet Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Juniper Ethernet Switch>
- **SNMP Port:** Port where Juniper Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring Juniper Ethernet Switch.

Monitoring Credentials

Choose the credential JSON file according to what SNMP credentials you'd like to use - SNMP v2c or SNMP v3.

SNMP V1/V2

- **Community String:** SNMPv2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Juniper Ethernet Switch JSON Files and Properties

Credential File: **omc_juniper_eth_switch_sample.json**

- **name**: Your Juniper Switch entity name.
- **displayName**: Your Juniper Switch entity display name.
- **timezoneRegion**: Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name**: Under "value", provide fully qualified host name or IP address of Juniper Switch
- **omc_dispatch_url**: Under "value", following the string snmp://, provide the fully qualified hostname or IP address of Juniper Switch.
- **omc_snmp_port**: Under "value", provide your SNMP port, default is 161.
- **omc_snmp_timeout**: Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version**: Under "value", provide the SNMP version used to monitor Juniper Switch, 2c by default.

Credential Files

omc_juniper_eth_switch_snmpv2c_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2.

- **COMMUNITY**: Use this credential file if you have configured your switch with SNMPv1/v2.

omc_juniper_eth_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3

- **authUser**: Use this credential file if you have configured your switch with SNMPv3. Under "value", within the square brackets, provide provide SNMPv3 user name.
- **authPwd**: Under "value", within the square brackets, provide the auth password or empty out the field.
- **authProtocol**: Under "value", within the square brackets, provide the auth-method (SHA or MD5).
- **privPwd**: Under "value", within the square brackets, provide the priv method password, if priv is used. Only the DES priv method is supported.

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Juniper Ethernet Switch, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Juniper MX Router

You can add Juniper MX Router entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Juniper MX Router for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during router configuration along with the IP address of the Cloud Agent which will be used for router monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the router.

Read-only access is adequate for MX router monitoring.

Step 2: Decide how you want to add the Juniper MX Router.

You can add Juniper MX Router entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Juniper MX Router Entity Type.
3. Enter the following UI properties.

Juniper MX Router UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Juniper MX Router>
- **SNMP Port:** Port where Juniper MX Router listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where the Juniper MX Router is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Juniper MX Router JSON Files and Properties

Definition File: `omc_juniper_mx_sample.json`

- **omc_dispatch_url:** snmp://<Fully qualified host name or IP address of Juniper MX Router>
- **omc_snmp_port:** Port where Juniper MX Router listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version:** SNMP version used to monitor Juniper MX Router (2c or 3) - 2c by default (optional)

Credential Files

`omc_juniper_mx_snmpv2_sample_creds.json`

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

`omc_juniper_mx_snmpv3_sample_creds.json`

SNMP v3

- **authUser:** SNMPv3 username.
 - **authPwd:** Password used for authentication.
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd:** Password used for encryption.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Juniper MX Router, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Juniper Netscreen Firewall

You can add Juniper Netscreen Firewall entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Juniper Netscreen Firewall for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during firewall configuration along with IP address of the Cloud Agent which will be used for Juniper firewall monitoring.

If SNMPv3 is used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) and password if authorization is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the firewall.

Read-only access is adequate for Juniper firewall monitoring.

Step 2: Decide how you want to add the Juniper Netscreen Firewall.

You can add Juniper Netscreen Firewall entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Juniper Netscreen Firewall Entity Type.
3. Enter the following UI properties.

Juniper Netscreen Firewall UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** `snmp://<Fully qualified host name or IP address of Juniper Netscreen Firewall>`
- **SNMP Port:** Port where Juniper Netscreen Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where Juniper Netscreen Firewall is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Juniper Netscreen Firewall JSON Files and Properties

Definition File: `omc_juniper_netscreen_sample.json`

- **omc_dispatch_url:** `snmp://<Fully qualified host name or IP address of Juniper Netscreen Firewall>`
- **omc_snmp_port:** Port where Juniper Netscreen Firewall listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout:** Timeout for SNMP requests in seconds - 30 seconds by default (optional)
- **omc_snmp_version:** SNMP version used to monitor Juniper Netscreen Firewall (2c or 3) - 2c by default (optional)

Credential Files

`omc_juniper_netscreen_snmpv2_sample_creds.json`

SNMP v2c

- **COMMUNITY:** SNMPv2c community string

`omc_juniper_netscreen_snmpv3_sample_creds.json`

SNMP v3

- **authUser:** SNMPv3 username
 - **authPwd:** Password used for authentication
 - **authProtocol:** Protocol used for authentication - supply either MD5 or SHA
 - **privPwd:** password used for encryption
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Juniper Netscreen Firewall, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Juniper SRX Firewall

You can add Juniper SRX Firewall entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Juniper SRX Firewall for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials are needed for monitoring.

If SNMPv1/v2 is used, you must supply the SNMP community string (which was entered during Juniper SRX Firewall configuration) along with IP address of agent that will be used to monitor the Juniper SRX Firewall.

If SNMPv3 is used, you must supply the SNMPv3 user, plus the authentication method (SHA or MD5) and authentication password, if authentication is used. In addition, privilege method (only DES supported) and privilege password will be required, if privileges are used. Everything must be manually configured up front in the Juniper SRX Firewall.

Read-only access is sufficient for Juniper SRX Firewall monitoring.

Step 2: Decide how you want to add the Juniper SRX Firewall.

You can add Juniper SRX Firewall entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Juniper SRX Firewall Entity Type.
3. Enter the following UI properties.

Juniper SRX Firewall UI Fields

- **Entity Name:** Name of your Juniper SRX Firewall in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Juniper SRX Firewall>
- **SNMP Port:** Port where Juniper SRX Firewall listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring your Juniper SRX Firewall.

Monitoring Credentials

SNMP V1/V2:

- **Community String:** SNMPv1/v2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Juniper SRX Firewall JSON Files and Properties

omc_juniper_srx_sample.json

- **name:** Your Juniper SRX Firewall entity name.
- **displayName:** Your Juniper SRX Firewall entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under “value”, provide fully qualified host name or IP address of the Juniper SRX Firewall.
- **omc_dispatch_url:** Under “value”, following the string snmp://, provide the fully qualified hostname or IP address of the Juniper SRX Firewall.
- **omc_snmp_port:** Under “value”, provide the port where the Juniper SRX Firewall listens for SNMP requests. The default is 161.
- **omc_snmp_timeout:** Under “value”, provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under “value”, provide the SNMP version used to monitor the Juniper SRX Firewall.

Credential Files

omc_juniper_srx_snmpv2_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2.

- **community:** Under “value”, within the square brackets, provide the SNMPv2c community string used during the Juniper SRX Firewall configuration.

omc_juniper_srx_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3.

- **authUser:** Under “value”, within the square brackets, provide the SNMPv3 username.
- **authPwd:** Under “value”, within the square brackets, provide the authorization password or empty out the field. .
- **authProtocol:** Under “value”, within the square brackets, provide the authorization method (SHA or MD5).
- **privPwd:** Under “value”, within the square brackets, provide the privilege method password, if privilege is used. Only the DES privilege method is supported.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Juniper SRX Firewall, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Kubernetes Cluster

You can add Kubernetes Cluster entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Kubernetes Cluster for monitoring.

For details on Kubernetes Cluster setup, see [Kubernetes Cluster](#).

Step 2: Decide how you want to add the Kubernetes Cluster.

You can add Kubernetes Cluster entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Kubernetes Cluster Entity Type.
3. Enter the following UI properties.

Kubernetes Cluster UI Fields

- **Discover Using Credentials:** Discover Kubernetes Cluster using Kubernetes Cluster credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Kubernetes Master URL:** Base URL of the API Server on the Kubernetes Master Node. The URL is of the form `http(s)://<hostname>:<port>`
- **Host Name:** Hostname of the Kubernetes master node
- **Heapster URL:** Base URL of Heapster. This needs to be specified if the performance metrics are to be collected from Heapster. If heapster is running inside Kubernetes as a cluster service the Base URL is of the form `http(s)://<host>:<port>/api/v1/namespaces/kube-system/services/heapster/proxy` Here, the host & port are same as in `omc_kubernetes_master_url`
- **Cloud Agent:** Cloud agent monitoring the host where the Kubernetes Cluster is installed.

Monitoring Credentials

Token Credentials

- **Token:** Token of the user going to discover Kubernetes
- **Keystore Certificate:** Certificate of Kubernetes API Server on Master Node. Users need to specify the text inside the certificate file if added from UI. In `omcli`, users need to create a Java Keystore, add certificate to that and specify the file path.
- **Certificate Alias:** Alias for the Certificate. This should be unique alphanumeric string
- **Trust Store Password:** Password of agent's Trust Store. This password is "welcome"

Basic Credentials

- **Username:** Username of the user going to discover Kubernetes
- **Password:** Password used for authentication.
- **Keystore Certificate:** Certificate of Kubernetes API Server on Master Node. Users need to specify the text inside the certificate file if added from UI. In `omcli`, users need to create a Java Keystore, add certificate to that and specify the file path.
- **Certificate Alias:** Alias for the Certificate. This should be unique alphanumeric string
- **Trust Store Password:** Password of agent's Trust Store. This password is "welcome"

Keystore Credentials

- **Store Location:** Location of Client keystore. This Java Keystore file (JKS) should contain client's certificate.
- **Store Type:** Store type. This value is always set to "JKS"
- **Store Password:** The keystore password to access the JKS file.
- **Keystore Certificate:** Certificate of Kubernetes API Server on Master Node. Users need to specify the text inside the certificate file if added from UI. In `omcli`, users need to create a Java Keystore, add certificate to that and specify the file path.
- **Certificate Alias:** Alias for the Certificate. This should be unique alphanumeric string
- **Trust Store Password:** Password of agent's Trust Store. This password is "welcome"

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Kubernetes Cluster JSON Files and Properties

Replace any text inside brackets <> excluding these brackets with your values according the legend within the brackets <>.

See [Kubernetes Cluster](#) for property descriptions.

Definition Files

omc_kubernetes_cluster_insecure.json

omc_kubernetes_cluster_secure.json

Credential Files

omc_kubernetes_cluster_basic_creds.json

omc_kubernetes_cluster_keystore_creds.json

omc_kubernetes_cluster_token_creds.json

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See [step 4. Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Kubernetes Cluster, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Microsoft IIS

You can add Microsoft IIS entities using the Add Entity UI or using the cloud agent command line interface (`omccli`) with the appropriate JSON files.

Step 1: Prepare Microsoft IIS for monitoring.

Prerequisites

Local Monitoring: Credentials are not required. The agent user is used for monitoring.

Remote Monitoring via WMI: Credentials are required. The credentials to be provided include the username and password used to log into the remote Windows host.

Before you can monitor Microsoft IIS entities, you must ensure the following prerequisites have been met:

- *Remote Monitoring of IIS:* If the Cloud agent and IIS are installed on different machines, then Microsoft Visual C++ needs to be installed on the Windows machine running the Cloud agent. The DLL `msvcr100.dll`, which is part of the Microsoft Visual C++ installation, is required.

Local Monitoring of IIS: If the Cloud agent and IIS are installed on the same machine, Microsoft Visual C++ is not required.

- IIS has been installed on a Windows Server. For more information about running the installation wizards from Server Manager, see [Installing IIS 8.5 on Windows Server 2012 R2](#).
- IIS Management Compatibility Components have been installed. To install the components:
 1. Click **Start**, click **Control Panel**, click **Programs and Features**, and then click **Turn Windows features on or off**.
 2. Follow the installation wizards and on the **Select Server Roles** page, select **Web Server (IIS)**. For more information about running the installation wizards from Server Manager, see [Installing IIS 8.5 on Windows Server 2012 R2](#).
 3. In Server Manager, expand Roles in the navigation pane and right-click Web Server (IIS), and then select Add Role Services.
 4. In the Select Role Services pane, scroll down to Web Server>Management Tools. Check the following boxes:
 - IIS 6 Management Compatibility
 - IIS 6 Metabase Compatibility
 - IIS 6 Management Console
 - IIS 6 WMI Compatibility
 5. Enable FTP Server.
- DCOM settings and WMI namespace security settings have been enabled for a remote WMI connection.

WMI uses DCOM to handle remote calls. DCOM settings for WMI can be configured using the DCOM Config utility (**DCOMCnfg.exe**) found in **Administrative Tools** in **Control Panel**. This utility exposes the settings that enable certain users to connect to the computer remotely through DCOM.

The following procedure describes how to grant DCOM remote startup and activation permissions for certain users and groups.

1. Click **Start**, click **Run**, type **DCOMCNFG**, and then click **OK**
2. In the **Component Services** dialog box, expand **Component Services**, expand **Computers**, and then right-click **My Computer** and click **Properties**

Prerequisites

3. In the **My Computer Properties** dialog box, click the **COM Security** tab
4. Under **Launch and Activation Permissions**, click **Edit Limits**
5. In the **Launch Permission** dialog box, follow these steps if your name or your group does not appear in the **Groups or user names list**
 - a. In the **Launch Permission** dialog box, click **Add**
 - b. In the **Select Users, Computers, or Groups** dialog box, add your name and the group in the **Enter the object names to select** box, and then click **OK**
6. In the **Launch Permission** dialog box, select your user and group in the **Group or user names** box. In the **Allow** column under **Permissions for User**, select **Remote Launch** and select **Remote Activation**, and then click **OK**

The following procedure describes how to grant DCOM remote access permissions for certain users and groups.

1. Click **Start**, click **Run**, type **DCOMCNFG**, and then click **OK**
2. In the **Component Services** dialog box, expand **Component Services**, expand **Computers**, and then right-click **My Computer** and click **Properties**
3. In the **My Computer Properties** dialog box, click the **COM Security** tab
4. Under **Access Permissions**, click **Edit Limits**
5. In the **Access Permission** dialog box, select **ANONYMOUS LOGON** name in the **Group or user names** box. In the **Allow** column under **Permissions for User**, select **Remote Access**, and then click **OK**

Allowing Users Access to a Specific WMI Namespace

It is possible to allow or disallow users access to a specific WMI namespace by setting the "Remote Enable" permission in the WMI Control for a namespace.

The following procedure sets remote enable permissions for a non-administrator user:

1. In the **Control Panel**, double-click **Administrative Tools**
2. In the **Administrative Tools** window, double-click **Computer Management**
3. In the **Computer Management** window, expand the **Services and Applications** tree and double-click the **WMI Control**
4. Right-click the **WMI Control** icon and select **Properties**
5. In the **Security** tab, select the namespace and click **Security**
6. Locate the appropriate account and check **Remote Enable** in the **Permissions** list

Firewall Settings

1. Click **Start**, click **Run**, type **GPEDIT.MSC**, and then click **OK**
 2. In the **Group Policy** dialog box, expand **Administrative Templates**, expand **Network**, expand **Network Connections**, and then expand **Windows Firewall**
 3. Select **Standard Profile** and double click on Windows Firewall : Allow Inbound Remote Administration Exceptions
 4. In the dialogue box that pops up, select **Enabled** and click on **Apply**
 5. If required, repeat the above 2 steps for **Domain Profile** as well
-

Step 2: Decide how you want to add the Microsoft IIS.

You can add Microsoft IIS entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Microsoft IIS Entity Type.
3. Enter the following UI properties.

Microsoft Internet Information Services UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Hostname of Microsoft IIS Server
- **Installation Directory:** Absolute installation path of the Microsoft IIS Server. You need to specify the path using double backslashes (\).
- **Logging Directory:** Absolute path to log file directory.
- **Cloud Agent:** Cloud agent monitoring the host where Microsoft Internet Information Services is installed.

Monitoring Credentials

- **Host Username:** Windows user on the Microsoft IIS Server host
- **Host Password:** Password for the Windows user

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Microsoft Internet Information Services JSON Files and Properties*Definition Files***omc_microsoft_iis_server_local_sample.json**

- **host_name:** Hostname of Microsoft IIS Server
- **install_dir:** Absolute installation path of the Microsoft IIS Server. You need to specify the path using double backslashes (\).
- **Example:** `C:\\Windows\\system32\\inetrv`

omc_microsoft_iis_server_remote_sample.json

- **omc_is_remote:** Property to indicate if the Microsoft IIS Server is local(no) or remote(yes)

Credential File: omc_microsoft_iis_server_remote_creds_sample.json

Credential properties (Applicable for remote monitoring via WMI)

- **wbem_username:** Windows user on the Microsoft IIS Server host
- **wbem_password:** Password for the Windows user

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE  
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Microsoft IIS, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Microsoft SCOM

You can add Microsoft SCOM entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Microsoft SCOM for monitoring.

Prerequisites

Credentials must follow the same criteria as any program which tries to obtain data from SCOM using the SCOM SDK. See [How to Connect an Operations Manager SDK Client to the System Center Data Access Service](#).

... The account that is used for authentication must be included in an Operations Manager user-role profile ...

The OMC Cloud Agent uses the `omc_scom.exe` client to connect to the SCOM SDK. The Cloud agent does not bundle required SCOM SDK libraries (due to the license type of libraries). You must manually copy the SCOM SDK libraries to the machine where the agent is running.

```
C:\Program Files\Microsoft System Center 2012 R2\Operations
Manager\Server\SDK Binaries\Microsoft.EnterpriseManagement.Runtime.dll
C:\Program Files\Microsoft System Center 2012 R2\Operations
Manager\Server\SDK
Binaries\Microsoft.EnterpriseManagement.OperationsManager.dll
C:\Program Files\Microsoft System Center 2012 R2\Operations
Manager\Server\SDK Binaries\Microsoft.EnterpriseManagement.Core.dll
```

Step 2: Decide how you want to add the Microsoft SCOM.

You can add Microsoft SCOM entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Microsoft SCOM Entity Type.
3. Enter the following UI properties.

SCOM (System Center Operations Manager) UI Fields

- **Entity Name:** Your SCOM entity name in Oracle Management Cloud.
 - **SCOM SDK Host:** Host name or IP address of an SCOM SDK Host.
 - **Cloud Agent:** Cloud agent monitoring the host where SCOM is installed.
- Monitoring Credentials (SCOM Credentials)
- **Username:** Username of the account which has access to SCOM..
 - **Password:** Password of the account which has access to SCOM.
 - **Domain:** Windows domain of the account which has access to SCOM.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using omcli and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

SCOM (System Center Operations Manager) JSON Files and Properties

Definition File: **omc_microsoft_scom_example.json**

- **name:** Your SCOM entity name.
- **displayName:** Your SCOM entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **omc_sdk_host:** Fully qualified host name or IP address of the server which hosts SCOM.

Credential File: **omc_microsoft_scom_creds.json**

- **username:** Username of the account which has access to SCOM..
 - **password:** Password of the account which has access to SCOM.
 - **domain:** Windows domain of the account which has access to SCOM.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Microsoft SCOM, see the following:

- [Lack of Data](#)

- [Create an Agent Support Bundle](#)

Add Microsoft SQL Server

You can add Microsoft SQL Server entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Microsoft SQL Server for monitoring.

Prerequisites

To enable monitoring for a Microsoft SQL Server Database, you can create a special database user as follows.

Create a user (for example, `moncs`) and map the new user to the `master` and `msdb` databases. Then, give this user the following minimum privileges.

 **Note:**

Beginning with Oracle Management Cloud 1.31, `sqladmin`-related privileges are no longer required.

```
CREATE LOGIN moncs
WITH PASSWORD = 'moncs';
GO
CREATE USER moncs FOR LOGIN moncs;
GO
```

Then, map the user `moncs`:

1. From the **Security** menu, select **Logins** `moncs`.
2. Right-click on `moncs` and select **Properties**.
3. Select **User Mapping**.
4. Map to all system and user databases:

```
USE master;
GRANT VIEW ANY DATABASE TO moncs;
GRANT VIEW ANY definition to moncs;
GRANT VIEW server state to moncs;
GRANT SELECT ON [sys].[sysaltfiles] TO [moncs];
GRANT execute on sp_helplogins to moncs;
GRANT execute on sp_readErrorLog to moncs;
```

```
GRANT EXECUTE ON dbo.xp_regread TO moncs;
```

```
USE msdb;
GRANT SELECT on dbo.sysjobsteps TO moncs;
GRANT SELECT on dbo.sysjobs TO moncs;
GRANT SELECT on dbo.sysjobhistory TO moncs;
```

For connecting to SQL server database with SSL encryption, do the following:

Prerequisites

1. Ensure the SQL server installation has the required updates for TLS 1.2 support as described in the following document.
<https://support.microsoft.com/en-in/help/3135244/tls-1-2-support-for-microsoft-sql-server>
2. Create a server certificate for the SQL server host.
Set up the certificate as mentioned in the section “Install a certificate on a server with Microsoft Management Console (MMC)” in the following document: <https://support.microsoft.com/en-in/help/316898/how-to-enable-ssl-encryption-for-an-instance-of-sql-server-by-using-mi>
3. Install the server certificate for the SQL server instance.
Set up the SQL server instance to use the server certificate created above, as mentioned in the section “To install a certificate for a single SQL Server instance” in the following document: <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/manage-certificates?view=sql-server-2016>
4. Export to a file, the root certification authority’s certificate that has signed the SQL server host certificate, and copy this file to the cloud agent host.
Export the certificate as described in section “Enable encryption for a specific client” in the following document: <https://support.microsoft.com/en-in/help/316898/how-to-enable-ssl-encryption-for-an-instance-of-sql-server-by-using-mi>
5. Create a trust store on the cloud agent host, and import the root certification authority’s certificate exported above.

```
keytool -import -file .\ca_cert.cer -alias mytrust -  
keystore .\trustStore.jks -storetype jks
```

6. Form the connection URL pointing to the trust store.

```
jdbc:sqlserver://  
xxx.xxx.com:1433;encrypt=true;trustServerCertificate=false;trustStore=C:\  
trustStore.jks;trustStorePassword=xxxx;
```

Step 2: Decide how you want to add the Microsoft SQL Server.

You can add Microsoft SQL Server entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Microsoft SQL Server Entity Type.
3. Enter the following UI properties.

Microsoft SQL Server Database UI Fields

- **Entity Name:** Name displayed in the Oracle Management Cloud console.
- **JDBC URL:** The connection URL for the MS SQL Server database. The URL follows the formats:

- Connect to default instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>
```

- Connect to named instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>\<Instance Name> (or)
```

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>;instanceName=<instance-name>
```

- Connect to instance by specifying port.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>:<SQL Server Database Port>
```

- Connecting with SSL encryption.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host Name>\<Instance Name>;encrypt=true;trustServerCertificate=false;trustStore=<Path to trust store file>;trustStorePassword=<trust store password>
```

See the prerequisites section for details on setting up the certificates and trust store.

- **Cloud Agent:** Agent monitoring the host on which the database is installed.
Monitoring Credentials
 - **Username:** MS SQL Server database user name to be used for monitoring.
 - **Password:** MS SQL Server database monitoring user's password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using omcli and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Microsoft SQL Server Database JSON Files and Properties

Definition File: **omc_sqlserver_db_sample.json**

- **name:** Your Microsoft SQL Server database name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **url:** Under "value", provide the connection URL for the MS SQL Server database. The URL follows the formats:
 - Connect to default instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host
Name>
```

- Connect to named instance.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host
Name>\\<Instance Name> (or)
jdbc:sqlserver://<Fully-qualified SQL Server Database Host
Name>;instanceName=<instance-name>
```

- Connect to instance by specifying port.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host
Name>:<SQL Server Database Port>
```

- Connecting with SSL encryption.

```
jdbc:sqlserver://<Fully-qualified SQL Server Database Host
Name>\\<Instance
Name>;encrypt=true;trustServerCertificate=false;trustStore=<
Path to trust store file>;trustStorePassword=<trust store
password>
```

See the prerequisites section for details on setting up the certificates and trust store.

Credential File: **omc_sqlserver_creds.json**

- **DBUserName:** Under "value", within the square brackets, provide the MS SQL Server database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
- **DBPassword:** Under "value", within the square brackets, provide the MS SQL Server database monitoring user's password.

Do not remove the square brackets.

-
3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Microsoft SQL Server, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add MongoDB

You can add MongoDB entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare MongoDB for monitoring.

Prerequisites

To enable monitoring for a MongoDB Database, you can create a special database user, for example, `omc_monitor` as follows:

1. Connect to your database:

```
use your MongoDB database name;
```

2. Create user:

```
db.createUser(  
  {  
    user: "omc_monitor",  
    pwd: "mongo123",  
    roles: [ "read" ]  
  }  
)
```

Step 2: Decide how you want to add the MongoDB.

You can add MongoDB entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the MongoDB Entity Type.

3. Enter the following UI properties.

MongoDB Database UI Fields

- **Entity Name:** Your MongoDB database name.
- **Host Name:** the fully-qualified host name where the MongoDB Database is installed.
- **Port:** MongoDB database port.
- **Database Name:** MongoDB database name.
- **Cloud Agent:** Cloud agent monitoring the host on which MongoDB is installed.

Monitoring Credentials

- **Username:** MongoDB database user name to be used for monitoring.
 - **Password:** MongoDB database monitoring user's password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

MongoDB Database JSON Files and Properties

Description File: `omc_mongodb_sample.json`

- **name:** Your MongoDB database name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under "value", provide the fully-qualified host name where the MongoDB Database is installed.
- **port:** Under "value", list the MongoDB database port.
- **database_name:** Under "value", list the MongoDB database name.

Credential File: `omc_mongodb_creds.json`

- **DBUserName:** Under "value", within the square brackets, provide the MongoDB database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
- **DBPassword:** Under "value", within the square brackets, provide the MongoDB database monitoring user's password.

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of MongoDB, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add MySQL Database

You can add MySQL Database entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare MySQL Database for monitoring.

Prerequisites

To enable monitoring for a MySQL Database, you can create a special database user, for example, `moncs` as follows:

1. Create a user:

```
CREATE USER 'moncs'@'l hostname' IDENTIFIED BY 'password';
```

2. Grant appropriate privileges:

```
GRANT SELECT, SHOW DATABASES ON *.* TO 'moncs'@'hostname' IDENTIFIED BY 'password';  
GRANT SELECT, SHOW DATABASES ON *.* TO 'moncs'@'%' IDENTIFIED BY 'password';
```

3. Flush privileges.
-

Step 2: Decide how you want to add the MySQL Database.

You can add MySQL Database entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the MySQL Database Entity Type.
3. Enter the following UI properties.

MySQL Database UI Properties

- **Entity Name:** Name displayed in the Oracle Management Cloud console.
 - **JDBC URL:** The connection URL for the MySQL Database. The URL follows the format: `jdbc:mysql://<host_name>:<port>/mysql` where `host_name` is a fully-qualified host name where MySQL Database is installed and `port` is the MySQL Database port defined at installation time.
 - **Host Name:** The fully-qualified host name where MySQL Database is installed.
 - **Cloud Agent:** Agent monitoring the host on which the database is installed.
- Monitoring Credentials
- **Username:** MySQL Database user name to be used for monitoring.
 - **Password:** MySQL Database user password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

MySQL Database JSON Properties and Files

Definition File: **omc_mysql_db_sample.json**

- **name:** Your MySQL database name.
- **display name:** Name displayed in the Oracle Infrastructure Monitoring Service User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: `America/New_York`
- **url:** Under "value", provide the connection URL used to connect to the installed MySQL database. The URL follows the format:
`jdbc:mysql://<host_name>:<port>/mysql` where `host_name` is a fully-qualified host name where MySQL Database is installed and `port` is the MySQL Database port defined at installation time.
- **hostname:** Under "value", provide the fully-qualified host name where MySQL Database is installed.
- **is_cluster:** (TRUE/FALSE) Specifies whether or not you are adding a MySQL Cluster Database.

Definition File: **omc_mysql_db_cluster_sample.json**

- **url**
`jdbc:mysql://host1:<port1>,host2:<port2>/dbname`
where:
 - Host 1 and Host 2 would be same in case of Single Host Cluster
 - Instance 1 / Node 1
instance_name: `<host1>.mycompany.com:<port1>`
 - Instance 2 / Node 2
instance_name: `<host2>.mycompany.com:<port2>`
- **jdbcdriver:** `com.mysql.jdbc.Driver`
- **MachineName:** Your MySQL Database Host Name
- **Is Cluster:** `true/false`
- **capability:** `monitoring`

Credential File: **omc_mysql_creds.json**

- **DBUserName:** Under "value", within the square brackets, provide the MySQL database user name to be used for monitoring. You must have defined this user in the Prerequisite Tasks step.
- **DBPassword:** Under "value", within the square brackets, provide the MySQL database monitoring user's password.

Do not remove the square brackets.

3. Add the entity using omcli.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of MySQL Database, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add NetApp FAS

You can add NetApp FAS entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Decide how you want to add the NetApp FAS.

You can add NetApp FAS entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the NetApp FAS Entity Type.
3. Enter the following UI properties.

NetApp FAS UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
 - **Host Name:** Fully qualified domain name (FQDN) or IP of the NetApp storage.
 - **SNMP Port:** Port to use for SNMP communication with NetApp storage.
 - **SNMP Timeout:** Timeout for SNMP communication with NetApp storage.
 - **SNMP Version:** Version of SNMP protocol to use for communication with NetApp storage.
 - **SNMP Community:** SNMP community string to use for communication with NetApp storage.
 - **Cloud Agent:** Cloud agent monitoring the host where NetApp FAS is installed.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

NetApp FAS JSON Files and Properties

Definition File: **omc_netapp_fas_sample.json**

- **host_name**: Fully qualified domain name (FQDN) or IP of the NetApp storage.
- **omc_snmp_port**: Port to use for SNMP communication with NetApp storage.
- **omc_snmp_timeout**: Timeout for SNMP communication with NetApp storage.
- **omc_snmp_version**: Version of SNMP protocol to use for communication with NetApp storage.
- **omc_snmp_community**: SNMP community string to use for communication with NetApp storage.

Credential File: **omc_netapp_fas_snmp_sample_creds.json**

- **authUser**: Name of a privileged user for SNMP communication.
 - **authPwd**: Password for a privileged user for SNMP communication.
 - **authProtocol**: Encryption protocol to be used for SNMP communication.
 - **privPwd**: Password for SNMP communication.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 2: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of NetApp FAS, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add NGINX

You can add NGINX entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Decide how you want to add the NGINX.

You can add NGINX entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the NGINX Entity Type.
3. Enter the following UI properties.

NGINX UI Fields

- **Entity Name:** Name of your NGINX entity in Oracle Management Cloud.
 - **Host Name:** Host where the NGINX server is running.
 - **Nginx Listen Port:** NGINX Server Port Number for connection to NGINX Status page.
 - **Nginx Binary File Path:** Full path to the NGINX binary file.
 - **Nginx PID File Path:** Full path to the NGINX PID file.
 - **Nginx Status Page URL:** URL used to access the NGINX status page.
 - **Cloud Agent:** Cloud agent monitoring the host where the NGINX server is installed.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

NGINX Files and Properties

Definition File: **omc_nginx.json**

- **host_name:** Host Name of the Nginx Target
 - **listen_port:** Nginx Server Port Number for connection to Nginx Status page
 - **install_home:** Nginx Server install directory
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of NGINX, see the following:

- [Lack of Data](#)

- [Create an Agent Support Bundle](#)

Add Oracle Access Manager/Oracle Internet Directory

You can add Oracle Access Manager/Oracle Internet Directory entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle Access Manager/Oracle Internet Directory for monitoring.

Prerequisites and Monitoring Credentials

The same credentials are used to discover the WebLogic Domain.

 **Note:**

Refresh of IDM targets is now supported. To refresh any IDM domain run `omcli refresh_entity agent ./idm_domain.json` where the content of `idm_domain.json` is:

```
{ "entities": [
  {
    "name": "Idm Domain",
    "type": "omc_weblogic_domain"
  }
]}
```

Step 2: Add the Oracle Access Manager/Oracle Internet Directory using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Internet Directory (OID)/Oracle Access Manager (OAM) JSON Files and Properties

Definition File: `omc_weblogic_domian.json`

- **displayName:** WebLogic Domain Entity Display Name that is displayed in the Infrastructure Monitoring UI time zone.
- **Region:** Time Zone (tz database time zones). For example: America/New_York.
- **port:** Port used for the WebLogic Admin Server(Console)
- **protocol:** The Protocol used for the WebLogic Server. For example: t3
- **admin_server_host:** Fully qualified WebLogic Admin Server Host Name where the WebLogic Admin Server is installed.

Credential File: `omc_weblogic_domain_creds.json`

- **user_name:** WebLogic Domain Entity User Name.
 - **password:** WebLogic Domain Entity Password.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Access Manager/Oracle Internet Directory, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Automatic Storage Management (ASM)

You can add Oracle Automatic Storage Management entities using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Note:

Oracle Management Cloud allows you to add all database components (DB, ASM, listener, etc.) as a single composite entity via the Oracle database system entity type. See [Add Oracle Database Systems](#).

Step 1: Prepare Oracle Automatic Storage Management for monitoring.

Credentials

Monitoring of ASM is supported through credential-based monitoring. For simplicity, use the default `asmnmp` user for the ASM monitoring credentials OR any user with both SYSASM and SYSDBA roles.

Step 2: Add the Oracle Automatic Storage Management using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Automation Storage Management JSON Files and Properties

Definition File: **omc_oracle_asm_sample.json**

- **name:** Your Oracle ASM entity name.
- **displayName:** Your Oracle ASM entity display name which is displayed on the Oracle Infrastructure Monitoring user interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **machine_name:** Under “value”, provide the fully-qualified machine name where the Oracle ASM is installed.
- **host_name:** Under “value”, provide the fully-qualified host name where the Oracle ASM is installed.
- **port:** Under “value”, list the Oracle ASM port.
- **sid:** Under “value”, list the Oracle ASM SID.

Credential File: **omc_oracle_asm_sample_creds.json**

- **user_name:** Under “value”, within the square brackets, provide the Oracle ASM user name to be used for monitoring.
- **password:** Under “value”, within the square brackets, provide the Oracle ASM monitoring user’s password.

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Automatic Storage Management, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Clusterware (CRS)

You can add Oracle Clusterware entities using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

 **Note:**

Oracle Management Cloud allows you to add all database components (DB, ASM, listener, etc.) as a single composite entity via the Oracle database system entity type. See [Add Oracle Database Systems](#).

Step 1: Prepare Oracle Clusterware for monitoring.**Prerequisite for Remote Monitoring**

:

SSH must be set up between the machine where the Cloud agent is installed and the machine where CRS is installed. The Cloud agent connects to the remote machine where CRS is installed via SSH authentication.

Step 2: Add the Oracle Clusterware using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Clusterware (CRS) JSON Files and Properties

Definition File: **omc_oracle_clusterware_sample.json**

- **scan_name** : Scan name for the cluster
- **cluster_name** : Cluster name
- **scan_port** : Scan port for the cluster
- **oracle_home** : CRS home base directory
- **omc_sshd_port** : SSH port value for remote monitoring
- **credential_ref** : "credentialRefs":["remote_sshcreds"] → for SSH Key based authentication

Credential Files

omc_oracle_clusterware_credless_sample.json

omc_oracle_clusterware_credential_sample.json

- **SSHUserName**: Your SSH user used to remotely log onto the listener host
- **SSHUserPassword** : Your SSH host Password. Optional , if there is a passwordless SSH setup. In this case, provide a private key field
- **SSH_PVT_KEY**: Path of your private key file. This private key is optional if the keys are generated at default location <user home>/ssh
- **sshdHost**: Your Cluster Host Name
- **sshdPort**: SSH port

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```


See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Clusterware, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Coherence Clusters

You can add Oracle Coherence Cluster entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle Coherence Cluster for monitoring.

Prerequisites

Supports both credential and non-credential monitoring. When using a secured JMX connection, a credential input file needs to be passed. For information on configuring a Coherence cluster, see [Configure a Coherence Cluster](#).

Step 2: Decide how you want to add the Oracle Coherence Cluster.

You can add Oracle Coherence Cluster entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle Coherence Cluster Entity Type.
3. Enter the following UI properties.

Oracle Coherence UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
 - **JMX Remote Port:** Coherence JMX port
 - **JMX Management Server Machine Name:** Coherence management node host.
 - **Do not discover caches:** If set to True, new Coherence cache targets will not be discovered. This is recommended for clusters with a very large number of caches (over 1000).
 - **Cloud Agent:** Cloud agent monitoring the host where Oracle Coherence is installed.
Monitoring Credentials (Coherence Credentials)
 - **Username:** JMX connection username.
 - **Password:** JMX connection password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Coherence JSON Files and Properties

Definition File: `omc_oracle_coherence.json`

- `omc_jmx_port` - Coherence JMX port
- `omc_machine_name` - Coherence management node host.
- `omc_skip_cache_discovery` - Specify that Coherence

Credential Files

`omc_oracle_coherence_cred.json`

`coherence_credentials.json`

If Coherence is configured using a secured JMX connection, then a credentials file has to be passed as an input argument.

- `omc_username` - JMX connection username.
 - `omc_password` - JMX connection password.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step [4. Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Coherence Cluster, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Database Listener Cluster

You can add individual Oracle Database Listener Cluster entities using the cloud agent command line interface (`omcli`) with the appropriate JSON files.



Note:

Oracle Management Cloud allows you to add all database components (DB, ASM, listener, etc.) as a single composite entity via the Oracle database system entity type. See [Add Oracle Database Systems](#).

Step 1: Add the Oracle Database Listener Cluster using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Database Listener Cluster JSON Files and Properties

Definition File: **omc_oracle_db_listener_cluster_sample.json**

- **displayName:** This is Oracle Database Listener Cluster Entity Display Name which is displayed on Infrastructure Monitoring UltimezoneRegion: Time Zone Example: PDT, GMT
- **host_name:** : Fully-qualified Host Name where the Oracle Database Listener Cluster is installed.
- **lsnr_alias** : Oracle Database Listener Cluster Alias
- **crs_home** : Absolute Path of the CRS HOME / GRID HOME

Credential Files

omc_oracle_db_listener_cluster_credless_sample.json

omc_oracle_db_listener_cluster_sample_cred.json

- **sshdPort:** SSHD Port on Remote host to Listen to Remote Cluster Listener
 - **SSHUserName:** SSH Host User Name
 - **SSHUserPassword:** SSH Host User Password
 - **SSH_PVT_KEY** : Location of the SSH private key copied from the remote machine where Cluster Listener is installed.
 - **SSH_PUB_KEY** : Location of the SSH public key copied from the remote machine where Cluster Listener is installed.
 - **sshdHost** : Host Name where Oracle Database Listener Cluster is installed.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 2: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Database Listener Cluster, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Database Listeners

You can add Oracle Database Listener entities using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Note:

Oracle Management Cloud allows you to add all database components (DB, ASM, listener, etc.) as a single composite entity via the Oracle database system entity type. See [Add Oracle Database Systems](#).

Step 1: Add the Oracle Database Listener using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Database Listener JSON Files and Properties

Definition File: **omc_oracle_db_listener_sample.json**

- **displayName:** This is Oracle Database Listener Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Fully-qualified Host Name where the Oracle Database Listener is installed.
- **port:** Oracle Database Listener port.
- **trace_dir_path:** Trace Log Files directory absolute path; optional parameter, define it if you are also using Oracle Log Analytics.
- **log_dir_path:** Alert Log Files directory absolute path; optional parameter, define it if you are also using Oracle Log Analytics.
- **Isnr_alias:** Oracle Database Listener Alias.

Credential Files:

omc_oracle_db_listener_local_credless.json**omc_oracle_db_listener_remote_ssh_sample.json****omc_oracle_db_listener_remote_ssh_sample_creds.json****omc_oracle_db_listener_creds.json**

- **displayName:** This is Oracle Database Listener Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **use_ssh:** Use Secure Shell protocol. (true/false).
True :- Set to true when performing remote monitoring.
False:- Set to false when performing local monitoring (agents 1.30 and lower). For agents 1.31 and greater, the use_ssh property is optional for local monitoring.
- **Isnr_port:** Your Oracle Database Listener Port.
- **Isnr_protocol:** Your Listener Protocol
- **oracle_home:** Your Oracle Listener ORACLE_HOME.
- **log_dir_path:** Alert Log Files directory absolute path; optional parameter, define it if you are also using Oracle Log Analytics.
- **trace_dir_path:** Trace Files directory absolute path.
- **Isnr_alias:** Your Oracle Database Listener Alias.
- **SSHUserName:** SSH host user name, on the host where the listener is installed.
- **SSHUserPassword:** SSH host user password
- **SSH_PVT_KEY:**Path of your private key file - Not required if a password is provided or SSH keys are available in the default location.
- **sshdPort:** SSH port.

 **Note:**

You must use a host user with SSH configured and enabled. Only password-based SSH is supported.

3. Add the entity using `omcli`.
Agent Local: Use `omc_oracle_db_listener_local_credless.json` as a template for the `DEFINITION_FILE`. No credentials required.

```
omcli add_entity agent DEFINITION_FILE
```

Agent Remote: Use `omc_oracle_db_listener_remote_ssh_sample.json` as a template for the `DEFINITION_FILE` with one of the credentials template files shown below for `CREDENTIAL_FILE`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

The following `CREDENTIAL_FILES` can be found in [Sample JSON files \(zip file\)](#)

- **omc_oracle_db_listener_cred_ssh.json:** Credentials using SSH password.
- **omc_oracle_db_listener_cred_ssh_pvtkey.json:** Credentials using SSH Private key.
- **omc_oracle_db_listener_credless_ssh.json:** Credentials using Credless SSH. Note: In order to use this file, you must first set up passwordless SSH between the agent host and the listener host.

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 2: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Database Listener, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Databases

You can add Oracle Database entities using the cloud agent command line interface (`omcli`) with the appropriate JSON files. Alternatively, these entities can be added as part of a Database System.

 **Note:**

Oracle Management Cloud allows you to add all database components (DB, ASM, listener, etc.) as a single composite entity via the Oracle database system entity type. See [Add Oracle Database Systems](#).

Step 1: Prepare Oracle Database for monitoring.

Prerequisites

Setting Up Monitoring Credentials for Oracle Database

Before you can begin monitoring DB systems, you must have the necessary privileges. A SQL script (`grantPrivileges.sql`) is available to automate granting these privileges. This script must be run as the Oracle DB SYS user. In addition to granting privileges, the `grantPrivileges.sql` script can also be used to create new or update existing monitoring users with the necessary privileges. For information about this SQL script, location and usage instructions, see [Creating the Oracle Database monitoring credentials for Oracle Management Cloud \(Doc ID 2401597.1\)](#).

Enabling TCPS Connections

Database Side (Single Instance)

1. Create the wallets.

```
mkdir -p /scratch/aime/wallets/rwallets
mkdir -p /scratch/aime/wallets/swallets
mkdir -p /scratch/aime/wallets/cwallets
```

2. To run the `orapki` commands go to the Oracle Home and run the following commands:

```
cd $ORACLE_HOME/bin

echo "***** Create Root wallet *****"

./orapki wallet create -wallet /scratch/aime/wallets/rwallets -
auto_login -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/rwallets -dn
"C=US,O=Oracle Corporation,CN=RootCA" -keysize 2048 -self_signed -
validity 365 -pwd oracle123 -addext_ski -sign_alg sha256

./orapki wallet export -wallet /scratch/aime/wallets/rwallets -dn
"C=US,O=Oracle Corporation,CN=RootCA" -cert /scratch/aime/wallets/
rwallets/cert.pem

./orapki wallet display -wallet /scratch/aime/wallets/rwallets

openssl x509 -noout -text -in /scratch/aime/wallets/rwallets/cert.pem

echo "***** Create server wallet *****"

./orapki wallet create -wallet /scratch/aime/wallets/swallets -
auto_login -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/swallets -trusted_cert
-cert /scratch/aime/wallets/rwallets/cert.pem -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/swallets -dn
"C=US,O=Oracle Corporation,CN=DBServer" -keysize 2048 -pwd oracle123 -
addext_ski -sign_alg sha256
```

Prerequisites

```
./orapki wallet export -wallet /scratch/aime/wallets/swallets -dn
"C=US,O=Oracle Corporation,CN=DBServer" -request /scratch/aime/wallets/
swallets/csr.pem

./orapki cert create -wallet /scratch/aime/wallets/rwallets -request /
scratch/aime/wallets/swallets/csr.pem -cert /scratch/aime/wallets/
swallets/cert.pem -validity 365 -sign_alg sha256 -serial_num $(date +
%s%3N)

./orapki wallet add -wallet /scratch/aime/wallets/swallets -user_cert -
cert /scratch/aime/wallets/swallets/cert.pem -pwd oracle123

openssl x509 -noout -text -in /scratch/aime/wallets/swallets/cert.pem

./orapki wallet display -wallet /scratch/aime/wallets/swallets

echo "***** Create client wallet *****"

./orapki wallet create -wallet /scratch/aime/wallets/cwallets -
auto_login -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/cwallets -trusted_cert
-cert /scratch/aime/wallets/rwallets/cert.pem -pwd oracle123

./orapki wallet add -wallet /scratch/aime/wallets/cwallets -dn
"C=US,O=Oracle Corporation,CN=DBClient" -keysize 2048 -pwd oracle123 -
addext_ski -sign_alg sha256

./orapki wallet export -wallet /scratch/aime/wallets/cwallets -dn
"C=US,O=Oracle Corporation,CN=DBClient" -request /scratch/aime/wallets/
cwallets/csr.pem

./orapki cert create -wallet /scratch/aime/wallets/rwallets -request /
scratch/aime/wallets/cwallets/csr.pem -cert /scratch/aime/wallets/
cwallets/cert.pem -validity 365 -sign_alg sha256 -serial_num $(date +
%s%3N)

./orapki wallet add -wallet /scratch/aime/wallets/cwallets -user_cert -
cert /scratch/aime/wallets/cwallets/cert.pem -pwd oracle123

openssl x509 -noout -text -in /scratch/aime/wallets/cwallets/cert.pem

./orapki wallet display -wallet /scratch/aime/wallets/cwallets
```

3. Change the mode of ewallet.p12.

```
chmod 666 /scratch/aime/wallets/swallets/ewallet.p12
chmod 666 /scratch/aime/wallets/cwallets/ewallet.p12
```

Listener Changes**Running SI on TCPS (Single Instance)**

Prerequisites

1. Create the Oracle Home.
2. Create a listener using TCP protocol (such as LIST).
3. Create a DB in the Oracle Home using the Listener created in Step 2. The Database and Listener might already be present.
4. Shut down the database instance.
5. Stop the Listener.

```
./lsnrctl stop LIST
```

6. Perform the following procedure.

Set the environment variables

```
export WALLET_LOCATION=/net/slc05puy/scratch/dbwallets
```

The wallet is already created and stored here. Make sure the wallet location is accessible from the current host.

```
export ORACLE_HOME=scratch/aimedb/12.1.0/12.1.0.2/dbhome_1
export ORACLE_SID=solssi
```

Back up the listener.ora, sqlnet.ora and tnsnames.ora files.

```
cp $ORACLE_HOME/network/admin/listener.ora $ORACLE_HOME/network/admin/
listener.ora.bckp
cp $ORACLE_HOME/network/admin/sqlnet.ora $ORACLE_HOME/network/admin/
sqlnet.ora.bckp
cp $ORACLE_HOME/network/admin/tnsnames.ora $ORACLE_HOME/network/admin/
tnsnames.ora.bckp
```

If sqlnet.ora is not present, create it.

```
touch $ORACLE_HOME/network/admin/sqlnet.ora
```

7. Modifying the ora files.

Listener.ora

Replace all 'TCP' with 'TCPS'

```
sed -i 's/TCP/TCPS/' $ORACLE_HOME/network/admin/listener.ora
```

Replace all '4343' with '2484' [43434 being the old listener port number]

```
sed -i 's/34343/2484/' $ORACLE_HOME/network/admin/listener.ora
```

Before executing the above shell commands, make sure you don't have any string other than the protocol which contains "TCP". This also applies

Prerequisites

to the for Listener port.

```
echo "SSL_CLIENT_AUTHENTICATION = TRUE" >> $ORACLE_HOME/network/admin/
listener.ora;
```

```
echo "WALLET_LOCATION =(SOURCE =(METHOD = FILE) (METHOD_DATA =(DIRECTORY
= $WALLET_LOCATION/swallets)))" >> $ORACLE_HOME/network/admin/
listener.ora;
```

```
echo "SSL_VERSION = 1.2" >> $ORACLE_HOME/network/admin/listener.ora;  **
Only if TLS version has to be 1.2
```

```
[SSL_VERSION = 1.2 or 1.1 or 1.0]
```

Sqlnet.ora

```
echo "SQLNET.AUTHENTICATION_SERVICES= (BEQ, TCPS)" >> $ORACLE_HOME/
network/admin/sqlnet.ora;
```

```
echo "SSL_CLIENT_AUTHENTICATION = TRUE" >> $ORACLE_HOME/network/admin/
sqlnet.ora;
```

```
echo "WALLET_LOCATION =(SOURCE =(METHOD = FILE) (METHOD_DATA =(DIRECTORY
= $WALLET_LOCATION/swallets)))" >> $ORACLE_HOME/network/admin/sqlnet.ora;
```

```
echo "SSL_VERSION = 1.2" >> $ORACLE_HOME/network/admin/sqlnet.ora;
```

```
** Only if TLS version has to be 1.2
```

8. Start the listener (./lsnrctl start LIST)
9. Start the database instance.
10. Run ./lsnrctl status LIST and check if the listener is running on TCPS with 2484 as the port and is associated with the database.

```
./lsnrctl status LTLS
LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 06-APR-2016
13:03:54
Copyright (c) 1991, 2014, Oracle. All rights reserved.
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS) (HOST=myhost.myco.com) (PORT=2484)))
```

```
STATUS of the LISTENER
```

```
-----
Alias                LTLS
Version              TNSLSNR for Linux: Version 12.1.0.2.0 -
Production
Start Date           06-APR-2016 10:41:33
Uptime               0 days 2 hr. 22 min. 21 sec
Trace Level          off
Security             ON: Local OS Authentication
SNMP                 OFF
```

```
Listener Parameter File
```

Prerequisites

```
/scratch/12102tls12/product/dbhome_1/network/admin/listener.ora
```

```
Listener Log File
```

```
/scratch/12102tls12/diag/tnslsnr/myhost/1tls/alert/log.xml
```

```
Listening Endpoints Summary...
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=myhost.myco.com)
(PORT=2484)))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC2484)))
```

```
Services Summary...
```

```
Service "sitls" has 1 instance(s).
```

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

```
Service "sitlsXDB" has 1 instance(s).
```

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

```
The command completed successfully.
```

You can see in the example that the database is now associated with the listener. If it is not, check whether the database `local_listener` parameter is set to the listener's connect descriptor.

```
alter system set local_listener='<CONNECT DESCRIPTOR FOR NEW LISTENER
PORT>';
```

Example: `alter system set local_listener='`

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS) (HOST=stra31.myco.com) (PORT=2484)))';
```

Once done, bounce the database instance. Even after doing this, if the database is not getting associated with the listener, and the listener is up and running without any issue, go to the ORACLE HOME and create a brand new database out of it using DBCA. It will prompt you to use the listener you just secured, and which is up and running on TCPS protocol.

TCPS Credentials

In order to establish secure communication with the Oracle Database, you must add TCPS Database Credential Properties to the credential JSON file in order to add the Oracle Database entity.

- **connectionTrustStoreLocation:** Your server/trust Key Store Location. This property is used to specify the location of the trust store. A trust store is a key store that is used when making decisions about which clients and servers can be trusted. The property takes a String value that specifies a valid trust store location.
- **connectionTrustStoreType:** Your server/trust Key Store Type. This property denotes the type of the trust store. It takes a String value. Any valid trust store type supported by SSL can be assigned to this property.
- **connectionTrustStorePassword:** Your server/trust Key Store Password. This property is used to set the password for the trust store. The trust store password is used to check the integrity of the data in the trust store before accessing it. The property takes a String value.
- **connectionKeyStoreLocation:** Your client Key Store Location. This property is used to specify the location of the key store. A key store is a database of key material that are used for various purposes, including authentication and data integrity. This property takes a String value.
- **connectionKeyStoreType:** Your client Key Store Type. This property denotes the type of the key store. It takes a String value. Any valid key store type supported by SSL can be assigned to this property.
- **connectionKeyStorePassword:** Your client Key Store Password. This property specifies the password of the key store. This password value is used to check the integrity of the data in the key store before accessing it. This property takes a String value.

Prerequisites

Agent Properties

Client authority

```
./omcli setproperty agent -name connectionKeyStoreLocation -value /scratch/  
aime/wallets/cwallets/ewallet.p12  
./omcli setproperty agent -name connectionKeyStoreType -value sha256  
./omcli setproperty agent -name connectionKeyStorePassword -value oracle123
```

Server authority

```
./omcli setproperty agent -name connectionTrustStoreLocation -value /  
scratch/aime/wallets/swallets/ewallet.p12  
./omcli setproperty agent -name connectionTrustStorePassword -value  
oracle123  
./omcli setproperty agent -name connectionTrustStoreType -value sha256
```

Once set, bounce the Agent.

```
./omcli stop agent  
./omcli start agent
```

 **Note:**

Make sure that the above wallet is accessible at the agent location.

Step 2: Add the Oracle Database using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Database System (single instance) JSON Properties and Files

Definition File: **omc_oracle_db_system_SI.json**

- **name**: Your Oracle Database Entity Name. Will also be used for the Database System name
- **displayName**: Your Oracle Database Entity Display Name. Will also be used for the Database System display name.
- **timezoneRegion**: Your timezone
- **omc_dbsys_config**: Configuration for DB System. Here, it is SI.
- **omc_dbsys_name_qualifier**: Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **host_name**: Name of the listener host that will be used to create the connect string to the database (host:port:SID or host:port:ServiceName)
- **omc_dbsys_port**: Listener port number used for connection requests
- **omc_dbsys_connect_type**: Specify type of connection: SID or Service Name
- **omc_dbsys_connect_value**: The value of the SID or Service Name
- **omc_dbsys_lsnr_alias**: Value of Listener Alias
- **omc_dbsys_home**: Oracle Home directory of the Listener
- **capability**: monitoring

Credential File: **omc_oracle_db_system_creds_SI_local.json**

- **DBUserName** : Your Database User Name
- **DBPassword** : Your Database Password
- **DBRole** : Your Database User Role. Default : Normal
- If Remote:

Credential File: **omc_oracle_db_system_creds_SI_with_SSH.json**

- **SSHUserName**: Your SSH user used to remotely logon to the listener host
 - **SSHUserPassword** : Your SSH host Password
 - **SSH_PVT_KEY**: Path of your private key file. This private key is optional if the keys are generated at default location <user home>/`.ssh`
 - **sshdPort**: SSH port
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Database, see the following:

- [Lack of Data](#)

- [Create an Agent Support Bundle](#)

Add Oracle Database Systems

An Oracle database system target is made up of components that constitute a logical database group. For example, it could be made up of a database and listener, or perhaps a database, listener, and ASM.

An Oracle database is typically dependent on other infrastructure entities such as a listener (for application connectivity) and ASM (for storage) in order for it to be available. Because these entities operate synergistically, Oracle Management Cloud allows you to add them as a single composite entity. This not only reduces the effort required to add them to Oracle Management Cloud, but also simplifies monitoring and managing them. Oracle Management Cloud accomplishes this via the *Oracle database system* entity type.

Single Instance Database System

When adding a database system for a single instance database, the following entities would be added:

- Oracle Database
- Pluggable Databases (if the DB is a Container)
- Database Instances
- Automatic Storage Management (ASM)
- Net Listener

RAC Database System

When adding a database system for a RAC database or RAC database with storage on ASM, the following entities would automatically be added.

- Oracle Database
- Pluggable Database (if the DB is a Container)
- SCAN Listener
- Database Instances
- ASM
- Oracle Clusterware

Availability

Since a database system is a logical grouping of entities required to make a database available to the connecting applications, the availability of a database system is determined from the status of its member entities: The database system is UP if all the member entities are also UP.

Shared Entities

Database system entities such as ASM, CRS, and SCAN listeners are *shared* entities such that they can be shared resources between multiple database systems. For example, four RAC database systems can have the same Oracle Clusterware entity.

Prerequisites

Setting Up Monitoring Credentials for Database System Discovery

You need to ensure that all monitoring credentials are set up before preparing to discover an Oracle database system. As a composite entity, all prerequisites and credentials for related entities need to be defined before the database system can be successfully added.

Before you can begin monitoring DB systems, you must have the necessary privileges. A SQL script (`grantPrivileges.sql`) is available to automate granting these privileges. This script must be run as the Oracle DB SYS user. In addition to granting privileges, the `grantPrivileges.sql` script can also be used to create new or update existing monitoring users with the necessary privileges. For information about this SQL script, location and usage instructions, see [Creating the Oracle Database monitoring credentials for Oracle Management Cloud \(Doc ID 2401597.1\)](#).

To monitor Oracle databases in OMC (using Infrastructure Monitoring or IT Analytics), the cloud agent requires database monitoring credentials, i.e. a database user with the appropriate set of privileges to collect metrics.

You can use either:

- The DBSNMP user (a user that is built-in with the Oracle Database)
OR
- You can create a new database user with appropriate privileges.

The DBSNMP user is provided as convenience since it is already predefined with all Oracle Databases. If you are using the DBSNMP user, it has sufficient privileges required for monitoring databases for Infrastructure Monitoring and IT Analytics. However, it also has additional privileges outside of what is required for Infrastructure Monitoring and IT Analytics.



Note:

To avoid the warning about a missing privilege on DBMS_LOCK, log in as `sysdba` and grant the following privilege to DBSNMP:

```
grant execute on sys.dbms_lock to DBSNMP
```

High Availability (Data Guard) and DBSNMP

Data Guard metrics are same as any other metric and can be collected using DBSNMP. But if the database is in MOUNTED state (eg. STANDBY database), not all metrics can be collected. Only Data Guard metrics can be collected and only by the SYSDBA monitoring user.

Review the following table for other prerequisites required for the other entity types.

Entity Type	Prerequisite
Oracle Database	Oracle Database (To configure Oracle Database with TCPS)

Entity Type	Prerequisite
Oracle Database Listener	If the listener host is remote from the Cloud agent, you need to set up SSH connectivity between the agent host and listener host.
Oracle Database Listener	
Oracle Database Cluster Listener	
Oracle Clusterware	
Oracle Cluster Node	

 **Note:**

When setting up RAC systems, the following user privileges must be used:

- *Local Monitoring:* The Agent user (OS user) should have privileges to run `olsnodes` and `srvctl` commands from GRID home.
- *Remote Monitoring:* The SSH user in SSH credentials should have privileges to run `olsnodes` and `srvctl` commands.

For SSH connectivity, you need to set the following JSON properties based on your chosen authentication option:

Authentication Options

Passwordless You need to set up passwordless SSH connectivity between the agent host and listener host.

- `SSHUserName`: The SSH user used to remotely log on to the listener host
- `sshdHost`: The Cluster Host Name
- `sshdPort`: The SSH port

Using a Password

- `SSHUserName`: The SSH user used to remotely log on to the listener host
- `SSHUserPassword`: The SSH host password. This parameter is optional *passwordless* SSH login is set up. When using *passwordless* SSH, you only need to provide a private key.
- `sshdHost`: The Cluster Host Name
- `sshdPort`: The SSH port

Using a Private Key

Entity Type	Prerequisite
Oracle Automatic Storage Management (ASM)	<ul style="list-style-type: none"> • SSHUserName: The SSH user used to remotely log on to the listener host • sshdHost: The Cluster Host Name • sshdPort: The SSH port • SSH_PVT_KEY: Path of your private key file. Specifying the private key parameter is optional if the keys are generated at default location <user home>/.ssh <p>Oracle Automatic Storage Management (ASM)</p>

Adding a Database System

1. Install the Cloud Agent on the Database Node.

The key advantage to adding a database system is that the Oracle database and all related entities are discovered from a single agent. The host itself is automatically discovered when the Cloud agent is installed. For RAC environments, or any multi-host database environment, you need to deploy a Cloud agent on each host in order to receive host metrics. For instructions on how to install agents, see [Install Cloud Agents](#).

2. Add the Oracle Database System.

Once your environment is set up, you are ready to add the Oracle database system entity. You can add a database system using the:

- Oracle Management Cloud Console
or
- Command Line Interface (OMCLI) via JSON files

Oracle Management Cloud Console

- a. From the Management Cloud console main menu, click **Administration**—>**Discovery**—>**Add Entity**. The Add Entity page displays.

- b. Select the Configuration (Single Instance or RAC) and enter the requisite configuration and monitoring credentials. UI parameters vary depending on the selected configuration.

 **Note:**

When setting up RAC systems, the following user privileges must be used:

- *Local Monitoring:* The Agent user (OS user) should have privileges to run `olsnodes` and `srvctl` commands from GRID home.
- *Remote Monitoring:* The SSH user in CRS credentials should have privileges to run `olsnodes` and `srvctl` commands.

Oracle Database System (single instance) UI Property Fields

- **Entity Name:** Your Oracle Database entity name. This name will also be used for the database system name
- **Configuration:** Configuration for database system: Single Instance or RAC
- **Name Prefix:** Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **Host Name:** Name of the listener host that will be used to create the connect string to the database (host:port:SID or host:port:ServiceName)
- **Port:** Listener port number used for connection requests
- **Connect Type:** Type of connection: SID or Service Name
- **Connect Value:** The value of the SID or Service Name
- **Listener Alias:** Value of Listener Alias
- **Listener Oracle Home:** Oracle Home directory of the Listener
The *Listener Oracle Home* field in the UI is the Oracle Home of the listener configured for that database. The Oracle Home for the listener may or may not be the same Oracle Home as the database as illustrated by the following example.

The following example shows two discrete database instances (prod_1 and test_1) in two separate Oracle Homes:

Oracle Home 1: /u01/app/oracle/product/19.0.0/prod_1

Oracle Home 2: /u01/app/oracle/product/19.0.0/test_1

Because both instances are configured with the listener in Oracle Home 1, to discover the test_1 instance (in Oracle Home 2) you would enter /u01/app/oracle/product/19.0.0/prod_1 in the *Listener Oracle Home* field.

- **Cloud Agent:** Cloud agent monitoring the database system.

Monitoring Credentials

- **Username:** Your Database User Name.
- **Password:** Your Database Password.
- **Database Role:** Your Database User Role (NORMAL/SYSDBA). Default is Normal.

SI with ASM (ASM Credentials)

- **Username:** Database user (ASM user name) that will be used by the cloud agent to connect to ASM.
- **Password:** Your ASM Password
- **Role:** Your ASM User role

Cloud Agent is not on the Cluster Host (Host SSH Credentials)

- **SSH Username:** Your SSH user used to remotely log on to the listener host.
 - **SSH Password:** Your SSH host Password.
 - **SSH Private Key:** Path of your private key file.
 - **SSH Port:** Your SSH port.
-

Oracle Database System (RAC) UI Fields

- **Entity Name:** Your Oracle Database Entity Name. Will also be used for the Database System name.
- **Name Prefix:** Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **SCAN Name:** Name of the SCAN Listener or SCAN VIP.
- **SCAN Port:** Port number on which the SCAN listener is listening for connections.
- **Service Name:** Service Name registered with the listener which is used to connect to the database.
- **Grid Home:** Oracle home directory for the Oracle Grid Infrastructure.
- **Cloud Agent:** Cloud agent used to monitor the cluster.

Monitoring Credentials

- **Username:** Your Database User Name.
- **Password:** Your Database Password.
- **Database Role:** Your Database User Role (NORMAL/SYSDBA). Default is Normal.

RAC with ASM (ASM Credentials)

- **Username:** Database user (ASM user name) that will be used by the cloud agent to connect to ASM.
- **Password:** Your ASM Password
- **Role:** Your ASM User role

Cloud Agent is not on the Cluster Host (Host SSH Credentials)

- **SSH Username:** Your SSH user used to remotely log on to the listener host.
- **SSH Password:** Your SSH host Password.
- **SSH Private Key:** Path of your private key file.
- **SSH Public Key:** Path of your public key file.
- **SSH Host Name:** Your Cluster Host Name.
- **SSH Port:** Your SSH port.

Command Line Interface (OMCLI) via JSON files

- a. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
- b. Edit the file(s) and specify the requisite properties for your chosen database system configuration.

Oracle Database System (single instance) JSON Properties and Files

Definition File: **omc_oracle_db_system_SI.json**

- **name:** Your Oracle Database Entity Name. Will also be used for the Database System name
- **displayName:** Your Oracle Database Entity Display Name. Will also be used for the Database System display name.
- **timezoneRegion:** Your timezone
- **omc_dbsys_config:** Configuration for DB System. Here, it is SI.
- **omc_dbsys_name_qualifier:** Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **host_name:**Name of the listener host that will be used to create the connect string to the database (host:port:SID or host:port:ServiceName)
- **omc_dbsys_port:** Listener port number used for connection requests
- **omc_dbsys_connect_type:** Specify type of connection: SID or Service Name
- **omc_dbsys_connect_value:** The value of the SID or Service Name
- **omc_dbsys_lsnr_alias:** Value of Listener Alias
- **omc_dbsys_home:** Oracle Home directory of the Listener
- **capability:** monitoring

Credential File: **omc_oracle_db_system_creds_SI_local.json**

- **DBUserName :** Your Database User Name
- **DBPassword :** Your Database Password
- **DBRole :** Your Database User Role. Default : Normal
- If Remote:

Credential File: **omc_oracle_db_system_creds_SI_with_SSH.json**

- **SSHUserName:** Your SSH user used to remotely logon to the listener host
 - **SSHUserPassword :** Your SSH host Password
 - **SSH_PVT_KEY:** Path of your private key file. This private key is optional if the keys are generated at default location <user home>/ .ssh
 - **sshdPort:** SSH port
-

Oracle Database System (RAC) JSON Properties and Files

Definition File: **omc_oracle_db_system_RAC.json**

- **name**: Your Oracle Database Entity Name. Will also be used for the Database System name
- **displayName**: Your Oracle Database Entity Display Name. Will also be used for the Database System display name.
- **timezoneRegion**: Your timezone.
- **omc_dbsys_config**: Configuration for DB System. Here, it is RAC.
- **omc_dbsys_name_qualifier**: Name that will be used to de-duplicate, if needed, the auto-generated names for the Listener and Cluster (SCAN) Listener. Generated name will be hostname-of-listener_Listener Alias
- **omc_db_system_scan_name**: Name of the SCAN Listener or SCAN VIP
- **omc_dbsys_port**: Port number on which the SCAN listener is listening for connections
- **omc_dbsys_connect_type**: Service Name
- **omc_dbsys_connect_value**: Service Name registered with the listener which is used to connect to the database
- **omc_dbsys_home**: Oracle home directory for the Oracle Grid Infrastructure
- **capability**: monitoring

Credential Files

omc_oracle_db_system_creds_RAC_local_with_ASM.json

omc_oracle_db_system_creds_RAC_with_SSH_with_ASM.json

omc_oracle_db_system_creds_RAC_local_without_ASM.json

omc_oracle_db_system_creds_RAC_with_SSH_without_ASM.json

Credential properties:

- **DBUserName** : Your Database User Name
- **DBPassword** : Your Database Password
- **DBRole** : Your Database User Role. Default : Normal

If ASM is also to be discovered:

- **user_name**: Your ASM User Name
- **password**: Your ASM Password
- **role**: Your ASM User role

If Remote:

- **SSHUserName**: Your SSH user used to remotely log onto the listener host
- **SSHUserPassword** : Your SSH host Password. Optional , if there is a passwordless SSH setup. In this case, provide a private key field
- **SSH_PVT_KEY**: Path of your private key file. This private key is optional if the keys are generated at default location <user home>/`.ssh`
- **sshdHost**: Your Cluster Host Name
- **sshdPort**: SSH port

-
- c. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREENTIAL_FILE [-encryption_method_gpg]]
```

- d. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

Deleting a Database System

When a database system is deleted, only the database system, the associated database and database's child entities (instances and pluggable databases) will be deleted. Other entities such as ASM, Clusterware and listeners will not be deleted.

Enabling Log Collection

Log collection via Log Analytics is available for specific types of database systems. To enable log collection, click the **Associate Logs** option. By default, the following logs are collected for the following entity types associated with the database system:

- Oracle Database Instance:
 - Database Trace Logs
 - Database Alert Logs
 - Database Incident Dump Files
- Oracle Database Clusterware
 - Clusterware Disk Monitor Logs
 - Clusterware Ready Services Alert Logs
 - Clusterware Ready Services Daemon Logs
- Oracle Database Listener
 - Database Listener Alert Logs
 - Database Listener Trace Logs
- Oracle ASM Instance
 - Automatic Storage Management Alert Logs
 - Automatic Storage Management Trace Logs

For more information about Log Analytics, see About Oracle Log Analytics.

Add Oracle ES2 Ethernet Switches

You can add Oracle ES2 Ethernet Switch entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle ES2 Ethernet Switch for monitoring.

Prerequisites

SNMPv1/v2 or SNMPv3 credentials needed for monitoring.

If SNMPv1/v2 is used, you must provide the SNMP community string that was entered during ES2 configuration along with IP address of the Cloud Agent which will be used for appliance monitoring.

If SNMPv3 used, you must provide the SNMPv3 user, plus authentication method (SHA or MD5) password if authentication is used, plus the privilege method (only DES is supported) and privilege password if privilege is used. All of this needs to be manually configured beforehand in the appliance.

Read-only access is adequate for the ES2 monitoring.

Step 2: Decide how you want to add the Oracle ES2 Ethernet Switch.

You can add Oracle ES2 Ethernet Switch entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle ES2 Ethernet Switch Entity Type.
3. Enter the following UI properties.

Oracle ES2 Ethernet Switch UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Dispatch URL:** `snmp://<Fully qualified host name or IP address of Oracle ES2 Ethernet Switch>`
- **SNMP Port:** Port where Oracle ES-2 Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring the host where the Oracle ES2 Ethernet Switch is installed.

Monitoring Credentials

SNMP V1/V2

- **Community String:** Community String for SNMP communication

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle ES2 Ethernet Switch JSON Files and Properties

Definition File: **omc_es2_sample.json**

- **omc_dispatch_url**: snmp://<Fully qualified host name or IP address of Oracle ES2 Ethernet Switch>
- **omc_snmp_port**: Port where Oracle ES-2 Ethernet Switch listens for SNMP requests - 161 by default (optional)
- **omc_snmp_timeout**: Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **omc_snmp_version**: SNMP version used to monitor Oracle ES-2 Ethernet Switch (2c or 3) - 2c by default (optional)

Credential Files

Choose the credential JSON file according to the SNMP version credentials you're using (SNMP v2c or SNMP v3).

omc_es2_snmpv2_sample_creds.json

SNMP v2c

- **COMMUNITY**: SNMPv2c community string.

omc_es2_snmpv3_sample_creds.json

SNMP v3

- **authUser**: SNMPv3 username.
 - **authPwd**: Password used for authentication.
 - **authProtocol**: Protocol used for authentication - supply either MD5 or SHA.
 - **privPwd**: Password used for encryption.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle ES2 Ethernet Switch, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle GoldenGate

You can add Oracle GoldenGate entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle GoldenGate for monitoring.

Prerequisites

Oracle GoldenGate enables the continuous, real-time capture, routing, transformation, and delivery of transactional data across heterogeneous (Oracle, DB2, MySQL, SQL Server, Teradata) environments. The following prerequisites apply when discovering and monitoring Oracle GoldenGate environments.

Enable Monitoring

The first prerequisite is to enable monitoring in GoldenGate. Follow the steps below for your specific GoldenGate version and architecture.

Classic Architecture

If you are using GoldenGate Classic Architecture, you will need to add a parameter in the GLOBALS file to enable monitoring.

You must be running GoldenGate version 12.3.0.1.181120 at a minimum. This is a cumulative patch set for GoldenGate released in Jan 9, 2019.

1. Locate the GLOBALS file in the top-level GoldenGate installation directory.
2. Add the following line to this file and save the file:

```
ENABLEMONITORING UDPPORT <port> HTTPPORT <port>
```

3. Restart GoldenGate Manager.

Microservices Architecture

If you are using GoldenGate Microservices Architecture, then as part of the setup of GoldenGate using the GoldenGate Configuration Assistant, you should enable Monitoring. Once monitoring has been enabled, the Performance Metric Server will be started. This is an indication that monitoring has been enabled for GoldenGate.

OCI GoldenGate

If you are using OCI GoldenGate, no prerequisites are required. Monitoring is enabled by default.

Import Certification for GoldenGate Secure Installations

If the Oracle GoldenGate setup is secure (HTTPS), the GoldenGate certificate needs to be imported into the agent manually prior to discovery. To do this, perform the following:

1. Extract the certificate from Oracle GoldenGate.

```
openssl s_client -showcerts -connect <hostname>:<service port>
```

2. Add the Oracle GoldenGate certificate to the cloud agent's JKS.

```
<jdk>/bin/keytool -importcert -file <goldengate-certificate> -  
alias goldengate -keystore <AGENT_HOME>/agent_inst/sysman/config/  
montrust/AgentTrust.jks -storepass welcome
```

Prerequisites

3. Bounce the cloud agent.

```
omcli stop agent ; omcli start agent
```

Step 2: Decide how you want to add Oracle GoldenGate.

You can add Oracle GoldenGate entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle GoldenGate Entity Type.
3. Enter the following UI properties.

Oracle GoldenGate UI Fields

 **Note:**

Credentials are required only for Oracle GoldenGate Microservice architecture.
No credentials are required for Oracle GoldenGate Classic architecture.

Classic

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **Port:** Oracle GoldenGate Service Manager port (if Microservice architecture). Performance Metric port if available, else Manager Port (if Classic architecture). Port to connect to OCI GoldenGate Service instance (if OCI GoldenGate architecture)
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where Oracle GoldenGate is installed.

Microservice

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **Service Manager Port:** Oracle GoldenGate Service Manager port (if Microservice architecture). Otherwise, Performance Metric port if available, else Manager Port (if Classic architecture)
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where Oracle GoldenGate is installed.

OCI GoldenGate

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **Port:** Port to connect to OCI GoldenGate Service instance, for example, 443.
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where Oracle GoldenGate is installed. The agent needs to be version 1.60 or higher.

Monitoring Credentials (Oracle GoldenGate Credentials)

- **Username:** Oracle GoldenGate Username
- **Password:** Oracle GoldenGate Password

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle GoldenGate JSON Files and Properties

 **Note:**

Credentials are required for both Oracle GoldenGate Microservice and Oracle GoldenGate OCI architectures. No credentials are required for Oracle GoldenGate Classic architecture.

Definition Files

omc_oracle_goldengate_sample_arch_classic.json

omc_oracle_goldengate_sample_arch_microservice.json

omc_oracle_goldengate_sample_arch_oci.json

- **host_name**: Fully-qualified Host Name where the Oracle GoldenGate is installed.
- **omc_port**: Oracle GoldenGate Service Manager port (if Microservice architecture). Performance Metric port if available, else Manager Port (if Classic architecture). Port to connect to OCI GoldenGate Service instance (if OCI GoldenGate architecture).
- **omc_ogg_arch**: Architecture - Microservice, Classic or OCI
- **omc_ogg_conn_timeout**: Connection Timeout in Seconds (Default 15 sec)

Credential File: **omc_oracle_goldengate_sample_creds.json**

Credentials (Microservice and OCI architecture only)

- **Alias**: Oracle GoldenGate Username
- **Password**: Oracle GoldenGate Password

-
3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Step 4: (Optional) Enable or Disable Log Collection for Oracle GoldenGate .

Optionally, you can use Log Analytics to monitor GoldenGate entity logs by associating them with the corresponding log sources. From the Discovery UI, click the **Associate Logs** option.

IMPORTANT: A local cloud agent (on the GoldenGate host) must be present before you can associate a GoldenGate entity with a log source. This agent must be monitoring the local OS (the OS should appear in the OMC UI). You can do this easily through the Discovery UI by specifying the cloud agent when adding the GoldenGate entity. Although you can add the GoldenGate entity without a cloud agent, log collection will not be enabled until an agent has been specified.

As with any Log Analytics integration, there needs to be an association between the Host and the entity for which you want to collect logs. For GoldenGate entities, this can be done easily through the Discovery UI. You enable log collection of the GoldenGate entities by associating them with the corresponding log sources. The Oracle GoldenGate entities and the corresponding log sources that you must associate them with are listed below:



Table C-1 GoldenGate Log Sources

GoldenGate Entity	Log Sources
GoldenGate Service Manager	Oracle GoldenGate Service Manager Logs
GoldenGate Performance Metric Server	Oracle GoldenGate Performance Metric Server Logs
GoldenGate Admin Server	Oracle GoldenGate Admin Server Logs
GoldenGate Distribution Server	Oracle GoldenGate Distribution Server Logs
GoldenGate Receiver Server	Oracle GoldenGate Receiver Server Logs
GoldenGate Deployment	Oracle GoldenGate Extract/Replicat Event Logs
GoldenGate Deployment	Oracle GoldenGate GGS Error Logs
GoldenGate Manager	Oracle GoldenGate Manager Report Files

You can change the association of log sources with entities, add more log sources as suitable for your application, or remove some from the list of log sources that are automatically enabled for collection. For information about associating entities with corresponding log sources, see *Work with Entity Associations in Using Oracle Log Analytics*.

Verifying Log Collection

Go to **Oracle Log Analytics Home** and verify the log collection. In case of error, you can take the following corrective actions:

- Ensure that the path of the logs in the log sources is correct. To change the location of the logs, or to provide an additional path from where the logs of a specific log source can be collected, do the following:
 1. From Oracle Log Analytics, click the OMC Navigation () icon on the top left corner of the interface. In the OMC Navigation bar, click **Log Admin**.
 2. In the **Log Sources** section, click the available count of log sources link.
 3. Click **Open Menu** () next to the log source entry that you want to edit and select **Edit**. For example, EBS Concurrent Manager Logs. The Edit Log Source page is displayed. In the **Included Patterns** tab, the default location of the logs is specified under the field **File Name Pattern**. For example, `{omc_oracle_goldengate_admin_servercsf}/`
`{omc_oracle_goldengate_admin_serverlog}/w*.mgr.`
 4. Click **Save**.
- Verify that the cloud agent user has access to the logs. Make the log files readable to the Oracle Management Cloud agents. See the section *Requirement for Logs Collection on Unix* in the topic *Generic Prerequisites for Deploying Oracle Management Cloud Agents in Installing and Managing Oracle Management Cloud Agents*.

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle GoldenGate, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle HTTP Server

You can add Oracle HTTP Server entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle HTTP Server for monitoring.

Prerequisites

OHS 12 : Node Manager credentials are required.

Also, the following prerequisites must be met:

- Cred-less (No credential file to be provided when running `omcli add_entity` during discovery) OHS discovery when the standalone OHS process owner and agent process owner are same user.
- Cred-based: OHS discovery when the standalone OHS process owner and agent process owners are different users.

 **Note:**

cred-less and cred-based discovery is applicable for standalone OHS 11. For OHS 12, only cred-based discovery is supported

- For HTTPS/Secured communication between OHS and the Cloud agent (for metric data collection) , the required certificate must be available with the agent in order for the SSL handshake to be successful. To make the certificate available with the agent :
 - Append the contents of your certificate file to the file : `/sysman/config/b64InternetCertificate.txt`
 - Ensure that only the following lines are appended to the `b64InternetCertificate.txt` file (that is, do not include blank lines, comments, or any other special characters):

```
-----BEGIN CERTIFICATE-----  
<<<Certificate in Base64 format>>>  
-----END CERTIFICATE-----
```

- Restart the agent by running the following commands :

```
omcli stop agent;omcli start agent;
```

Step 2: Decide how you want to add the Oracle HTTP Server.

You can add Oracle HTTP Server entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle HTTP Server Entity Type.
3. Enter the following UI properties.

Oracle HTTP Server (OHS) UI Fields

- **Entity Name:** Name of your Oracle HTTP Server.
- **Host Name:** Host Name of the Oracle HTTP Server
- **Oracle Home:** Absolute path of the Oracle Home.
- **Instance Home(11g) / Domain Home:** Absolute path of the Instance Home (11g)/ Absolute path of the Domain Home (12c and later)
- **Component Name:** Oracle HTTP Server component name.
- **Version:** Oracle HTTP Server installed version.
- **Configuration Path:** httpd.conf file directory path.
- **Listen Port:** Port of the Oracle HTTP Server.
- **Protocol:** Protocol used to connect to the Oracle HTTP Server. (HTTP/HTTPS)
- **Cloud Agent:** Cloud agent monitoring the host where Oracle HTTP Server is installed.

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle HTTP Server (OHS) Files and Properties

Definition File: **omc_oracle_apache_sample.json**

- **host_name:** Host Name of the Oracle HTTP Server
- **port:** Port of the Oracle HTTP Server
- **ohs_home:** Absolute path of the Instance Home (11g)/ Absolute Path of the Domain Home (12c)
- **component_name:** Component Name
- **protocol:** Protocol for connecting to the Oracle HTTP Server
- **config_path:** httpd.conf file directory path - file name not to be appended.
- **oracle_home:** Absolute path of the Oracle Home
- **version:** Version of OHS installed.

Credential Files

omc_oracle_apache_sample_creds_ohs12.json

Use this credential JSON file if you are running OHS 12

- **nm_user_name:** Node Manager username
- **nm_password:** Node Manager password

omc_oracle_apache_sample_creds_ohs11.json

Use this credential JSON file if you are running OHS 11 (optional)

- **HostUserName:** Host username
 - **HostPassword:** Host password
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE  
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle HTTP Server, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Identity Manager

You can add Oracle Identity Manager entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle Identity Manager for monitoring.

Prerequisites

Same credentials used to discover the WebLogic Domain.

 **Note:**

Refresh of IDM targets is now supported. To refresh any IDM domain run `omcli refresh_entity idm_domain.json` where the content of `idm_domain.json` is:

```
{ "entities": [  
  {  
    "name": "Idm Domain",  
    "type": "omc_weblogic_domain"  
  }  
]}
```

Step 2: Decide how you want to add the Oracle Identity Manager.

You can add Oracle Identity Manager entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle Identity Manager Entity Type.
3. Enter the following UI properties.

Oracle Identity Manager (OIM) UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Port:** Port used for the WebLogic Admin Server(Console)
- **Protocol:** Protocol used for the WebLogic Server. For example: t3
- **Administration Server Host:** Fully-qualified WebLogic Admin Server Host Name where the WebLogic Admin Server is installed.
- **Discover Coherence:** Discover Oracle Coherence.
- **Cloud Agent:** Cloud agent monitoring the host where OIM is installed.

Monitoring Credentials (WebLogic Server Credentials)

- **Username:** WebLogic Domain Entity User Name.
- **Password:** WebLogic Domain Entity Password.

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Identity Manager (OIM) JSON Files and Properties

Definition File: **omc_weblogic_domian.json**

- **displayName:** WebLogic Domain Entity Display Name that is displayed in the Infrastructure Monitoring UI time zone.
- **Region:** Time Zone (tz database time zones). For example: America/New_York.
- **port:** Port used for the WebLogic Admin Server(Console)
- **protocol:** Protocol used for the WebLogic Server. For example: t3
- **admin_server_host:** Fully-qualified WebLogic Admin Server Host Name where the WebLogic Admin Server is installed.

Credential File: **omc_weblogic_domain_creds.json**

- **user_name:** WebLogic Domain Entity User Name.
- **password:** WebLogic Domain Entity Password.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Identity Manager, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Infiniband Switch

You can add Oracle Infiniband Switch entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle Infiniband Switch for monitoring.

Prerequisites

To enable monitoring, HTTP and SNMPv1/v2c/v3 are needed.

If SNMPv1/v2 is used, you must provide SNMP community string that has been used earlier in IB Switch configuration along with IP address of agent which will be used for IB Switch monitoring.

If SNMPv3 is used, in addition to SNMPv3 user, you must provide the auth method (SHA or MD5) and auth-password if auth used, and plus priv method (only DES supported) and priv-password if priv used. You must configure everything manually in IB Switch. Read only access is sufficient for IB Switch monitoring.

Step 2: Decide how you want to add the Oracle Infiniband Switch.

You can add Oracle Infiniband Switch entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle Infiniband Switch Entity Type.
3. Enter the following UI properties.

Oracle Infiniband Switch UI Fields

- **Entity Name:** Name of your Oracle Infiniband Switch in Oracle Management Cloud.
- **Dispatch URL:** snmp://<Fully qualified host name or IP address of Oracle InfiniBand Switch>
- **SNMP Port:** Port where Oracle InfiniBand Switch listens for SNMP requests - 161 by default (optional)
- **SNMP Timeout:** Timeout for SNMP requests in seconds - 30 secs by default (optional)
- **Cloud Agent:** Cloud agent monitoring your Oracle Infiniband Switch.

Monitoring Credentials

SNMP V1/V2

- **Community String:** SNMPv2c community string.

SNMP V3

- **Username:** SNMPv3 username.
 - **Authorization Password:** Password used for authentication
 - **Authorization Protocol:** Protocol used for authentication. (MD5 or SHA)
 - **Privacy Password:** password used for encryption
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle InfiniBand Switch JSON Files and Properties

Definition File: **omc_oracle_ib_switch_sample.json**

- **name:** Your InfiniBand Switch entity name.
- **displayName:** Your InfiniBand Switch entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under "value", provide fully qualified host name or IP address of InfiniBand Switch
- **omc_dispatch_url:** Under "value", following the string snmp://, provide the fully qualified hostname or IP address of InfiniBand Switch.
- **omc_snmp_port:** Under "value", provide your SNMP port, default is 161.
- **omc_snmp_timeout:** Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under "value", provide the SNMP version used to monitor InfiniBand Switch, 2c by default.

Credential Files

omc_oracle_ib_switch_snmpv2c_sample_creds.json

Use this credential file if you have configured your switch with SNMPv1/v2.

- **COMMUNITY:** Under "value", within the square brackets, provide the SNMPv2c community string used during the InfiniBand Switch configuration.

omc_oracle_ib_switch_snmpv3_sample_creds.json

Use this credential file if you have configured your switch with SNMPv3

- **authUser:** Under "value", within the square brackets, provide provide SNMPv3 user name.
- **authPwd:** Under "value", within the square brackets, provide the auth password or empty out the field.
- **authProtocol:** Under "value", within the square brackets, provide the auth-method (SHA or MD5).
- **privPwd:** Under "value", within the square brackets, provide the priv method password, if priv is used.

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle InfiniBand Switch, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle JVM Runtime

You can add Oracle JVM Runtime entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle JVM Runtime for monitoring.

Prerequisites

Monitoring Oracle JVM Runtime can be performed in the following modes:

1. No user authentication, No SSL
2. No user authentication, SSL
3. User authentication, No SSL
4. User Authentication, SSL

SSL configuration:

You will need to import your truststore certificate into the cloud agent truststore using `omcli` as shown in the following example:

```
$ omcli secure add_trust_cert_to_jks -alias <Alias of cert to import>
-trust_certs_loc <Cert file to import>
```

Step 2: Decide how you want to add the Oracle JVM Runtime.

You can add Oracle JVM Runtime entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle JVM Runtime Entity Type.
3. Enter the following UI properties.

Oracle JVM Runtime UI Properties

- **Discover Using Credentials:** Discover JVM Runtime using JVM credentials (on by default).
- **Entity Name:** Name of the JVM Runtime entity appearing in the UI.
- **Host Name:** The host where the JVM application is running.
- **JMX Port Number:** The JMX port where the JVM application is running.
- **Cloud Agent:** The cloud agent monitoring the JVM application.

Monitoring Credentials

- **JMX Remote Access Username:** JVM username.
 - **JMX Remote Access Password:** JVM password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle JVM Runtime JSON Files and Properties

Definition File: `omc_jvm_sample_creds.json`

- **JVM host name (`hostname`)** - the host where the JVM application is running. It is specified during discovery in the discovery JSON or discovery UI. The value populated during discovery is the value specified by the user in discovery parameters
- **JVM application JMX port (`omc_jmx_port`)** - the JMX port where the JVM application is running. It is specified during discovery in the discovery JSON or discovery UI. The value populated during discovery is the value specified by the user in discovery parameters
- **JVM runtime name (`omc_runtime_name`)** - value determined by querying runtime MBean value Name (. The format of the value is `<pid>@hostname`.
- **JVM application JMX service URL (`omc_jmx_service_url`)** - This values is calculated from the typical remote JMX service URL format -

```
service:jmx:rmi:///jndi/rmi://<hostname>:<jmx_port>/jmxrmi
```

where the hostname and jmx port is determined from the values specified by the user in the discovery parameters.

- **JVM application class name - (`omc_jvm_application_class_name`)** - value determined by querying runtime MBean value SystemProperties. The value is located by the key "sun.java.command" in the list of system properties.
- **JVM application instance command line arguments (`omc_jvm_app_command_line_args`)** - value determined by querying runtime MBean value InputArguments. The value returned by the MBean is formatted from an array to space delimited string. This is the command line arguments specified by the user when the JVM application is started

Identifying properties

The identifying properties specified in the target model for reconciliation purposes are the following:

- `host_name`
- `omc_jmx_port`
- `omc_jvm_application_class_name`

Credential File: `omc_jvm_sample_creds.json`

- **user_name:** JVM username.
 - **password:** JVM password.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle JVM Runtime, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle NoSQL Database

You can add Oracle NoSQL Database entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle NoSQL Database for monitoring.

Credentials

Monitoring of Oracle NoSQL is supported only through credential-less JMX (no credentials JSON file is needed).

Step 2: Decide how you want to add the Oracle NoSQL Database.

You can add Oracle NoSQL Database entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle NoSQL Database Entity Type.
3. Enter the following UI properties.

Oracle NoSQL UI Fields

- **Entity Name:** Your Oracle NoSQL Database entity name.
 - **Host Name:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
 - **URL:** The NoSQL database connection URL.
 - **Port:** The NoSQL database port.
 - **Store Name:** Oracle NoSQL Database store name.
 - **NoSQL Java Home:** Java home of NoSQL database or any Java home of version above 1.8 .
 - **KV Home Location:** KV home of the NoSQL database.
 - **Cloud Agent:** Agent monitoring the host on which the database is installed.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle NoSQL JSON Files and Properties

Definition File: **omc_nosql_db_sample.json**

- **name:** Your Oracle NoSQL Database entity name.
- **displayName:** This is Oracle NoSQL Database Entity Display Name which is displayed on Infrastructure Monitoring UI.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under "value", provide the fully-qualified host name where the Oracle NoSQL Database is installed.
- **omc_port:** Under "value", list the NoSQL Database port.
- **omc_url:** Connection URL to connect to the installed Oracle NoSQL database. Comma separated host:port of all Nosql nodes of a store. example: <host_name>:<port>,<host_name>:<port>
- **omc_nosql_java_home:** Under "value", list the Java home of NoSQL database or any Java home of version above 1.8 .
- **omc_kv_home:** Under "value", list the KV home of the NoSQL database.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle NoSQL Database, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Pluggable Database

You can add Oracle Pluggable Database entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 2: Add the Oracle Pluggable Database using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Pluggable Database (PDB) (Standalone) JSON Properties and Files

Definition File: `omc_oracle_pdb_sample.json`

- **displayName:** This is Oracle Pluggable Database (PDB) Entity Display Name which is displayed on Infrastructure Monitoring UI
- **timezoneRegion:** Time Zone Example: PDT, GMT etc
- **host_name :** Fully-qualified Host Name for the Oracle Pluggable Database (PDB)
- **omc_pdb_tbsp_port:** Oracle Pluggable Database (PDB) port
- **omc_pdb_tbsp_service_name:** Oracle Pluggable Database (PDB) Service Name

Credential File: `omc_oracle_pdb_cred_sample.json`

- **DBUserName:** Oracle Pluggable Database (PDB)username
 - **DBPassword:** Oracle Pluggable Database (PDB) user's password
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Pluggable Database, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Power Distribution Unit (PDU)

You can add PDU entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare PDU for monitoring.

Prerequisites

To enable monitoring, HTTP and SNMPv1/v2c/v3 are needed. The NMS and trap tables in PDU administration interface must be set for a proper SNMP monitoring. For more information, see the PDU vendor documentation.

Step 2: Add the PDU using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Power Distribution Unit (PDU) Files and Properties

Definition File: `omc_oracle_pdu_sample.json`

- **name:** Your PDU entity name.
- **displayName:** Your PDU entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: `America/New_York`
- **omc_dispatch_url:** Under "value", provide your PDU HTTP URL.
- **omc_snmp_port:** Under "value", provide your SNMP port, default is 161.
- **omc_snmp_timeout:** Under "value", provide the timeout for SNMP requests in seconds, 10 by default.
- **omc_snmp_version:** Under "value", provide the SNMP version used to monitor PDU. Valid values are 1, 2 or 3.

Credential Files

You need to use HTTP credentials along with one of the SNMP credentials (v2c or v3)

HTTP credentials - part of `SNMPv1` and `SNMPv3 json`

- **username:** User name for HTTP communication.
- **password:** Password for user in HTTP communication.

SNMPv2c

`omc_oracle_pdu_sample_snmpv1_creds.json`

- **COMMUNITY:** Community String for SNMP communication

SNMP v3

`omc_oracle_pdu_sample_snmpv3_creds.json`

Use this credentials JSON file if using SNMP v3.

- **authUser:** Name of privileged user for SNMP communication
- **authPwd:** Password for privileged user for SNMP communication
- **authProtocol:** Encryption protocol to be used for SNMP communication
- **privPwd:** Password for SNMP communication

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of PDU, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Service Bus

You can add Oracle Service Bus entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle Service Bus for monitoring.

Prerequisites

Important: Before you can monitor Oracle Service Bus (OSB) entities in Oracle Management Cloud, you must first enable monitoring from the Oracle Service Bus Administration console.

ORACLE[®] Service Bus 11gR1

The screenshot shows the Oracle Service Bus Administration console. On the left, there is a 'Change Center' window with 'weblogic session' selected and buttons for 'Activate', 'Discard', and 'Exit'. Below it is the 'Project Explorer' showing a tree view of projects: 'default', 'J2eeApp', 'BusinessServices', 'ProxyServices', 'v2', 'v3', and 'wsdls'. The main area displays 'View a Proxy Service (J2eeApp/ProxyServices/v2/CustomPS)'. A table shows metadata: 'Last Modified By' (weblogic), 'Last Modified On' (1/21/19 2:42 AM), 'References' (2 Ref(s)), and 'Referenced By' (0). Below the table are tabs for 'Configuration Details', 'Operational Settings', 'SLA Alert Rules', and 'Policies'. The 'Operational Settings' tab is active, showing 'General Configuration' with 'State' set to 'Enabled'. Under 'Monitoring', 'Enable Pipeline Monitoring' is checked and set to 'Action' level or above. 'Aggregation Interval' is set to 0 hours and 1 minute. 'Enable Alerting' is checked and set to 'Normal' level or above.

For information, see [What are Operational Settings for a Service?](#)

Once monitoring has been enabled from the Oracle Service Bus Administration console, you can add OSB entities to Oracle Management Cloud. When specifying an OSB entity, you use credentials of a user with at least the **Monitor** role. The user can also have either the **Operator** or **Admin** role.

Step 2: Add the Oracle Service Bus using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Prerequisites

Important: Before you can monitor Oracle Service Bus (OSB) entities in Oracle Management Cloud, you must first enable monitoring from the Oracle Service Bus Administration console.

ORACLE Service Bus 11gR1

The screenshot shows the Oracle Service Bus Administration console interface. On the left is the 'Project Explorer' showing a tree view of projects including 'J2eeApp', 'BusinessServices', 'ProxyServices', 'v2', 'v3', and 'wsdlis'. The main area displays the configuration for a 'Proxy Service (J2eeApp/ProxyServices/v2/CustomPS)'. The 'Monitoring' section is expanded, showing the following settings:

Section	Setting	Value
General Configuration	State	Enabled
Monitoring	Enable Pipeline Monitoring at	Action level or above
	Aggregation Interval	0 hours 1 mins
	SLA Alerts	Enable Alerting at Normal level or above

For information, see [What are Operational Settings for a Service?](#)

Once monitoring has been enabled from the Oracle Service Bus Administration console, you can add OSB entities to Oracle Management Cloud. When specifying an OSB entity, you use credentials of a user with at least the **Monitor** role. The user can also have either the **Operator** or **Admin** role.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Service Bus, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Traffic Director

You can add Oracle Traffic Director entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle Traffic Director for monitoring.

Prerequisites

OTD 11

Use an OTD Administrator user.

In addition, to enable collection of metrics, you must configure and start an SNMP subagent. To start the SNMP subagent, use OTD Admin Console, or use the following command:

```
tadm start-snmp-subagent
--host=<otd_host>
--port=<otd_port>
--user=<otd user>
--password-file=<password_file>
```

For more information on configuring and starting an SNMP subagent, see the Oracle Traffic Director documentation.

OTD 12

Use a WebLogic Server user with the Monitor role. The user can also have Operator or Admin roles, which include the Monitor role.

Step 2: Decide how you want to add the Oracle Traffic Director.

You can add Oracle Traffic Director entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle Traffic Director Entity Type.
3. Enter the following UI properties.

Traffic Director Instance (OTD) UI Fields

12c or later

- **Entity Name:** The OTD entity name that is displayed on Infrastructure Monitoring UI.
- **Port:** OTD Administration Server listening port
- **Protocol:** t3 or t3s
- **Administration Server Host:** Fully-qualified host name where the OTD Administration Server is installed.
- **Discover Coherence:** Coherence Clusters deployed on the domain will be discovered. Turn this option off when discovering a SOA Suite Domain.
- **Cloud Agent:** Cloud agent monitoring the host where the OTD Administration Server is installed.

Monitoring Credentials (WebLogic Server Credentials)

- **Username:** OTD Administration Server username.
- **Password:** OTD Administration Server password.

11g

- **Entity Name:** OTD 11g Entity Display Name which is displayed on Infrastructure Monitoring UI
- **Administration Server Host Name:** Fully-qualified Host Name where the OTD 11g Administration Server is installed.
- **Administration Server Listen Port:** OTD 11g Administration Server listening port.
- **Configuration Name:** OTD 11g configuration name
- **Administration Server Oracle Home:** Absolute path of the OTD 11g Administration Server oracle home location
- **Cloud Agent:** Cloud agent monitoring the host where the OTD 11g Administration Server is running.

Monitoring Credentials (OTD 11g Administration Server Credentials)

- **Administration Username:** OTD 11g Administration Server username
- **Administration Password:** OTD 11g Administration Server password
- **SNMP Community String:** Community string used in SNMP aubagent. Default value is "public".

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Traffic Director Instance (OTD) JSON Files and Properties

Definition File: **omc_oracle_otd_cluster_sample.json** **Note:**

This applies specifically for OTD 11. For OTD 12, OTD will be automatically discovered as part of the WebLogic Domain discovery

- **name:** Your OTD 11g name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **admin_host:** Under “value”, provide the fully-qualified Host Name where the OTD 11g Administration Server is installed.
- **admin_port:** Under “value”, provide the OTD 11g Administration Server listening port.
- **admin_oracle_home:** Under “value”, provide the absolute path of the OTD 11g Administration Server Oracle Home location.
- **config_name:** Under “value”, provide the OTD 11g configuration name.

Credential File: **omc_oracle_otd_cluster_sample_creds.json** **Note:**

This applies specifically for OTD 11. For OTD 12, OTD will be automatically discovered as part of the WebLogic Domain discovery.

- **admin_user_name:** Under “value”, within the square brackets, provide the OTD 11g Administration Server user name. You must have identified this user in the Prerequisite Tasks step.
- **admin_password:** Under “value”, within the square brackets, provide the OTD 11g Administration Server password.
- **snmp_comm_string:** Under “value”, within the square brackets, provide the community string used in SNMP subagent. The default value is "public".

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Traffic Director, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Unified Directory

You can add Oracle Unified Directory entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle Unified Directory for monitoring.

Prerequisites

- i) OUD Gateway
- ii) OUD Replication
- iii) OUD Proxy

LDAP username and LDAP passwords are used to connect to the OUD LDAP server.

OID Credentials:

Directory Server Username and Password: The username and password that will be used by the agent to bind to the server instance. Ensure the password is in the appropriate field.

The following credential JSON sample illustrates how the properties should be entered.

```
{ "entities":[
  {
    "name":"OMC_OUD_Directory1",
    "type":"omc_oud_directory",
    "displayName":"OUD_directory1",
    "timezoneRegion":"PST",
    "credentialRefs":["OudCreds"],
    "properties":{
      "host_name":{"displayName":"Directory Server
Host","value":"myserver.myco.com"},
      "omc_ldap_port":{"displayName":"Administration
Port","value":"4444"},
      "omc_trust_all":{"displayName":"Trust ALL Server SSL
certificates","value":"true"},
      "capability":
{"displayName":"capability","value":"monitoring"}}
    }
  ]}

{"credentials":[
  {
    "id":"OudCreds","name":"OUD Credentials","credType":"MonitorCreds",
    "properties":[{"name":"authUser", "value":"CLEAR[cn=Directory
Manager]"},
                  {"name":"authPasswd", "value":"CLEAR[mypassword]"}]
  }
]}
```

Step 2: Add the Oracle Unified Directory using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.

2. Edit the file(s) and specify the requisite properties shown below.

Oracle Unified Directory JSON Files and Properties

- i) OUD Server
- ii) OUD Proxy Server

*Definition Files***omc_oud_directory.json****omc_oud_proxy.json**Credential File: **omc_oud_creds.json**

Replace any text inside brackets <> excluding these brackets with your values according the legend inside <>

- **Administration Port:** The administration port of the target server instance.
- **Directory Server Host:** The fully qualified domain name of the target server instance. For replicated servers, you must provide the same host name that was used when replication was configured.
- **Trust All :** Set to true by default. This implies that all the certificates that are presented by the server (or servers, in the case of replication) will be accepted automatically. Change this setting if you want to specify different behavior. (Optional) If you have changed the default setting for the Trust All field, enter a path in the Trust Store Path field..

The agent will use the trust store located in this path to validate the certificates of the administration connector that are presented by the server(s). This path must be readable by the agent (and thus located in a file system that is accessible by the agent). The trust store must contain the public keys of the administration connector certificates. It must be in JKS format and must not be password protected.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Unified Directory, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle Virtual Networking

You can add Oracle Virtual Networking entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Decide how you want to add the Oracle Virtual Networking.

You can add Oracle Virtual Networking entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle Virtual Networking Entity Type.
3. Enter the following UI properties.

Oracle Fabric Manager / Virtual Networking / Xsigo UI Fields

- **Entity Name:** Your OFM/OVN entity name.
 - **Host Name:** The fully qualified host name or IP address of the OFM/OVM host.
 - **REST Port:** The OFM/OVN REST port, default is 8443.
 - **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
 - **Cloud Agent:** Cloud agent monitoring the host where OFM/OVM is installed.
Monitoring Credentials
 - **Username:** The OFM/OVN user.
 - **Password:** The OFM/OVM user password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle Fabric Manager / Virtual Networking / Xsigo JSON Files and Properties

 **Note:**

ONLY proper SSL certificates of OVN/OFM are supported. For self-signed certificates, manual addition to the agent keystore is required. To manually add a self-signed certificate to the agent keystore, run the following command:

```
omcli secure add_trust_cert_to_jks -password  
<ask_oracle_support> -trust_certs_loc </path/to/  
certificateOfOFMServer.crt> -alias <hostname_of_OFM>
```

omc_oracle_ovn_sample.json

- **name:** Your OFM/OVN entity name.
- **displayName:** Your OFM/OVN entity display name.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **omc_ovn_hostname:** Under “value”, provide fully qualified host name or IP address of the OFM/OVM host.
- **omc_ovn_rest_port:** Under “value”, provide your OFM/OVN REST port, default is 8443.

Credential File: **omc_oracle_ovn_sample_creds.json**

- **username:** Under “value”, provide fully qualified host name or IP address of the OFM/OVM host.
- **password:** Under “value”, within the square brackets, provide the OFM/OVN user password.

Do not remove the square brackets.

-
3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 2: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle Virtual Networking, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle VM Manager

You can add Oracle VM Manager entities using the Add Entity UI or using the cloud agent command line interface (omcli) with the appropriate JSON files.

Step 1: Prepare Oracle VM Manager for monitoring.

Prerequisites

The cloud agent must be deployed on the Oracle VM Manager host.

Credentials: The username and password are required to access the Oracle VM Manager console.

Example:

```
username=admin / password=admin_pw
```

Certificates:

You need to explicitly add the Oracle VM Manager Weblogic certificate to the Agent's JKS.

How to extract certificate from Oracle VM Manager:

To export the Oracle VM Manager WebLogic certificate, log in as the root user and enter the following command:

```
#!/u01/app/oracle/ovm-manager-3/ovm_upgrade/bin/ovmkeytool.sh
exportca >
    <file_loc_for_certificate>
```

To import the Oracle VM Manager Weblogic certificate to the Agent Keystore, log in as an Oracle cloud agent user and enter the following command:

```
<AGENT_INSTANCE_HOME>/bin/omcli secure          add_trust_cert_to_jks -
trust_certs_loc
    <file_loc_for_certificate> -alias <alias_name>
```

Step 2: Decide how you want to add the Oracle VM Manager.

You can add Oracle VM Manager entities from the UI or using OMCLI and entity JSON files.

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select an Oracle VM Manager Entity Type.
3. Enter the following UI properties.

Oracle VM Manager UI Properties

- **Entity Name:** Name appearing in the UI.
 - **Oracle VM Manager Console URL:** Oracle VM Manager console URL used to connect to the installed Oracle VM Manager. The URL follows the format: `https://<ovm_host_name>:<port>/ovm` where `ovm_host_name` is a fully-qualified host name where Oracle VM Manager is installed and `port` refers to the port number on which Oracle VM Manager is listening.
 - **Cloud Agent:** The cloud agent monitoring the Oracle VM Manager application.
Monitoring Credentials
 - **Admin Username:** Oracle VM Manager username.
 - **Admin Password:** Oracle VM Manager password.
-

See [Add Entities from the Console](#) for more information.

Using omcli and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle VM Manager JSON Files and Properties

Definition File: **omc_oracle_vm_manager.json**

- **name:** Your Oracle VM Manager name. This needs to be unique across OVM Managers used
- **display name:** Name displayed in the Oracle Infrastructure Monitoring Service User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example:

```
America/New_York
```

- **omc_ovmm_console_url:** Under "value", provide the Oracle VM Manager console URL used to connect to the installed Oracle VM Manager. The URL follows the format: `https://<ovm_host_name>:<port>/ovm` where `ovm_host_name` is a fully-qualified host name where Oracle VM Manager is installed and `port` refers to the port number on which Oracle VM Manager is listening.

Credential File: **omc_oracle_vm_manager_creds.json**

- **OVMUsername:** Under "value", within the square brackets, provide the Oracle VM Manager console user name (default admin) to be used for monitoring.
- **OVMPassword:** Under "value", within the square brackets, provide the Oracle VM Manager console monitoring user's password.

Do not remove the square brackets.

3. Add the entity using OMCLI.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Apache HTTP Server, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle VM Server for SPARC (LDOMS)

You can add Oracle VM Server entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle VM Server for monitoring.**Prerequisites**

- Prerequisites: OMC Cloud Agent is deployed on the LDoms Control Domain
- Discovery does not require any user credentials but you need to grant *solaris ldoms read RBAC* privileges to the OMC Cloud Agent user:

```
/usr/sbin/usermod -A solaris.ldoms.read oracle
```

- Discovery properties:
 - The following command retrieves the LDoms Control Domain UUID to be supplied at discovery time through entity identifying property *omc_virtual_platform_id* using *virtinfo*:

```
# virtinfo -ap | grep DOMAINUUID
DOMAINUUID|uuid=280c9ff4-a134-48cd-cee9-a270b2aaefa0
```
- Autodiscovery of LDoms-related entities:
 - Use a JSON file with details to discover the Oracle VM Server for SPARC (LDoms). Using this method, all Logical Domains (Virtual Machines) are automatically discovered and updated periodically when things change in the Oracle VM Server for SPARC (LDoms) deployment.

Step 2: Add the Oracle VM Server using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Oracle VM Server for SPARC (LDoms) JSON Files and Properties

Definition File: **omc_sparc_ldoms_sample.json**

- **omc_virtual_platform_id**: LDoms Control Domain UUID
- **omc_virtual_type**: LDoms
- **omc_dispatch_url**: local://localhost

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file  
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle VM Server, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Oracle WebLogic Server/Domain

You can add Oracle WebLogic Server/Domain entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Oracle WebLogic Server/Domain for monitoring.

Prerequisites

To enable monitoring of a Oracle WebLogic Server (WLS), use a WebLogic user with at least the Monitor role. The user can also have Operator or Administrator roles, which include the Monitor role. If you have enabled the Oracle WebLogic Server with SSL, you must export the certificate from its keystore and import it in the Cloud Agent keystore. Perform the following steps:

1. Stop the Cloud Agent.

```
omcli stop agent
```

2. Export the certificate from the WLS instance JMX SSL keystore to the Cloud Agent's truststore. For example, on a UNIX host:

```
cd <agent base Directory>/agentStateDir/sysman/config/montrust
keytool -exportcert -alias <alias of WLS SSL key> -file <Exported Cert Name> -keystore <path to the WLS SSL Keystore>.keystore -storepass <WLS SSL Keystore password> -rfc
```

3. Import the WLS instance JMX SSL keystore to the Cloud Agent's truststore:

```
keytool -import -noprompt -alias <alias agent's truststore key> -file <Exported Cert Name>.cer -keystore AgentTrust.jks -storepass <Agent truststore password, default is "welcome">
```

4. Restart the Cloud Agent..

```
omcli start agent
```

Step 2: Decide how you want to add the Oracle WebLogic Server/Domain.

You can add Oracle WebLogic Server/Domain entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Oracle WebLogic Server/Domain Entity Type.
3. Enter the following UI properties.

WebLogic Domain /WebLogic Server UI Fields

- **Entity Name:** Your WebLogic Domain name.
 - **Port:** The port used for WebLogic Admin Server(Console).
 - **Protocol:** The protocol used for WebLogic Server - For example: t3
 - **Administration Server Host:** the fully-qualified host name where the WebLogic Admin Server is installed.
 - **Discover Coherence:** (True/False) Specify whether Coherence Clusters deployed on the Weblogic domain should be discovered. This option is set to True by default. Turn this option off when discovering a SOA Suite domain.
 - **Cloud Agent:** Cloud agent monitoring the host where WebLogic is installed.
Monitoring Credentials (WebLogic Server Credentials)
 - **Username:** WebLogic Server user name.
 - **Password:** WebLogic Server username password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

WebLogic Domain /WebLogic Server JSON Files and Properties

Definition File: **omc_weblogic_domain_sample.json**

 **Note:**

When you add a WebLogic Domain entity (requiring credentials), because Oracle Management Cloud connects to the WebLogic Admin Server, all WebLogic Clusters and WebLogic Servers that are part of that domain are automatically discovered. There's no need to add them separately.

- **name:** Your WebLogic Domain name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **port:** Under "value", provide the port used for WebLogic Admin Server(Console).
- **protocol:** Under "value", provide the protocol used for WebLogic Server - For example: t3
- **admin_server_host:** Under "value", provide the fully-qualified host name where the WebLogic Admin Server is installed.

Credential File: **omc_weblogic_domain_sample_creds.json**

- **username:** Under "value", within the square brackets, provide the WebLogic Domain monitoring user name. You must have defined this user in the Prerequisite Tasks step.
- **password:** Under "value", within the square brackets, provide the WebLogic Domain monitoring user's password.

Do not remove the square brackets.

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Oracle WebLogic Server/ Domain, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add SPARC/Intel Computers

You can add SPARC/Intel Computers entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare SPARC/Intel Computers for monitoring.

Credentials

Only the username and password are required to use SSH to log in to the ILOM service processor.

Step 2: Add the SPARC/Intel Computers using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Intel/SPARC Computers JSON Files and Properties

Definition File: **omc_ilom_server.json**

- **omc_dispatch_url**: ilom-ssh://<fully qualified host name or IP address of ILOM Server>

Credential File: **omc_ilom_server_creds.json**

- **username**: ILOM Server username (use Administrator role).
 - **password**: ILOM Server password (use Administrator role).
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE  
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of SPARC/Intel Computers, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add Tomcat

You can add Tomcat entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare Tomcat for monitoring.

Prerequisites and Credentials

Tomcat is monitored using JMX. You must configure Tomcat for JMX remote monitoring even if you are using a local agent.

Tomcat can be monitored with or without authentication. If a JMX credential is created, then it's assumed you're monitoring this entity with credentials.

To create a JMX credential for monitoring:

1. Edit the environment file:

```
vi $CATALINA_HOME/bin/setenv.sh
```

Add:

```
CATALINA_OPTS="-Dcom.sun.management.jmxremote -  
Dcom.sun.management.jmxremote.port=9999 -  
Dcom.sun.management.jmxremote.ssl=false -  
Dcom.sun.management.jmxremote.authenticate=true -  
Dcom.sun.management.jmxremote.password.file=../conf/jmxremote.password -  
Dcom.sun.management.jmxremote.access.file=../conf/jmxremote.access"
```

2. Save the file.
3. Change the file permission as executable:

```
chmod 755 $CATALINA_HOME/bin/setenv.sh
```

4. Edit the password file:

```
vi $CATALINA_HOME/conf/jmxremote.password
```

Add:

```
control tomcat  
admin tomcat
```

5. Edit the access file:

```
vi $CATALINA_HOME/conf/jmxremote.access
```

Add:

```
control readonly  
admin readwrite
```

Prerequisites and Credentials

6. Change the file permission for only the owner:

```
chmod 600 jmxremote.access
chmod 600 jmxremote.password
```

7. Bounce the Tomcat instance:

```
sh $CATALINA_HOME/bin/shutdown.sh
sh $CATALINA_HOME/bin/startup.sh
```

If you have enabled the Tomcat JMX with SSL, you must export the certificate from its keystore and import it in the Cloud Agent keystore. Perform the following steps:

1. Export the certificate from the Tomcat instance JMX SSL keystore to the Cloud Agent's truststore. For example, on a UNIX host:

```
cd <agent Base Directory>/agentStateDir/sysman/config/montrust
keytool -exportcert -alias <alias of Tomcat JMX SSL key> -file <Exported
Cert Name>.cer -keystore <path to the Tomcat JMX SSL Keystore>.keystore -
storepass <Tomcat JMX SSL Keystore password> -rfc
```

2. Import the Tomcat instance JMX SSL keystore to the Cloud Agent's truststore:

```
keytool -import -noprompt -alias <alias agent's truststore key> -file
<Exported Cert Name>.cer -keystore AgentTrust.jks -storepass <agent
truststore password, default is "welcome">
```

3. Restart the agent, using the command line interface:

```
omcli stop agent
omcli start agent
```

Step 2: Decide how you want to add the Tomcat.

You can add Tomcat entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the Tomcat Entity Type.
3. Enter the following UI properties.

Tomcat UI Fields

- **Discover Using Credentials:** Discover Tomcat using Tomcat credentials (on by default).
- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **Host Name:** The fully-qualified host name where the Tomcat entity is installed.
- **JMX Port Number:** The JMX port used by the Tomcat entity.
- **Cloud Agent:** The cloud agent monitoring the host where Tomcat is installed.

Monitoring Credentials

- **JMX Username:** The Tomcat user name.
 - **JMX Password:** The Tomcat user name password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

Tomcat JSON Files and Properties

Definition File: **omc_tomcat_secure_sample.json**

- **name:** Your Tomcat name.
- **displayName:** Name displayed in the Oracle Infrastructure Monitoring User Interface.
- **timezoneRegion:** Time zone of your entity. It is recommended that you use the long values IANA-maintained TZ database time zones. For example: America/New_York
- **host_name:** Under "value", provide the fully-qualified Tomcat host name where the Tomcat entity is installed.
- **jmx_port:** Under "value", provide the JMX port used by the Tomcat entity.

Credential Files

omc_tomcat_secure_creds.json

- **jmx_username:** Under "value", within the square brackets, provide the Tomcat user name. Leave this field blank and still include the credential JSON file for credential-less discovery. Do not remove the square brackets.
- **jmx_password:** Under "value", within the square brackets, provide the Tomcat user name password. Leave this field blank and still include the credential JSON file for credential-less discovery.

omc_tomcat_secureSSL_creds.json

- **ssl_trust_store:** Under "value", within the square brackets, provide the full path to the Cloud Agent truststore, AgentTrust.jks. For example, <agent base directory>/sysman/config/montrust/AgentTrust.jks
- **ssl_trust_store_password:** Under "value", within the square brackets, provide the Cloud Agent truststore password, the default is "welcome".

Do not remove the square brackets.

 **Note:**

To add a Tomcat entity that does not require credentials, simply add the entity without any credentials. And, if you do not provide any credentials, make sure input JSON file also does not contain any references to credentials.

To add a Tomcat entity without credentials, you will still need to provide the credentials file (omc_tomcat_secure_creds.json) but keep the `jmx_username` value blank, as shown in the following example.

```
{
  "credentials": [{
    "id": "TomcatCredsNormal",
    "name": "tomcat_creds",
    "credType": "TomcatCreds ",
    "properties": [{
      "name": "jmx_username",
      "value": "CLEAR[]"
    }, {
      "name": "jmx_password",
      "value": "CLEAR[]"
    }
  ]
}]
}
```

Tomcat JSON Files and Properties

 **Note:**

Beginning with Oracle Management Cloud 1.30, Tomcat discovery will always use the Agent Trust Store. User-provided SSL Trust Store will no longer be accepted.

-
3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of Tomcat, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add VMware vCenter

You can add VMware vCenter entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Prepare VMware vCenter for monitoring.

Prerequisites

In order for the Cloud Agent to be able to collect all the metrics for the Oracle Management Cloud VMware entities, you should:

1. Install VMware tools on the VM host.
2. Set the statistics level to one (1).

Credentials: username/password required to access VMware vCenter (use Administrator role).

Example:

```
username=Administrator@vsphere.local / password=<admin_pw>
```

Certificates:

You need to explicitly add the vCenter certificate to the Agent's JKS:

Example:

```
<jdk>/bin/keytool -importcert -file <vmware-vsphere-certificate> -alias  
vmware -keystore $T_WORK/agentStateDir/sysman/config/montrust/  
AgentTrust.jks -storepass welcome
```

How to extract certificate from vCenter:

```
openssl s_client -showcerts -connect <hostname>:443
```

Discovery properties:

How to retrieve VMware vCenter Server Instance UUID to be passed in at discovery time through the entity property `omc_virtual_mgmt_system_id` using VMware PowerCLI:

Example:

```
PS C:\> $vcenter = Connect-viserver vcsa-01a.corp.local -User  
Administrator@vsphere.local -Password admin_pw  
PS C:\> $vcenter.InstanceUuid  
d322b019-58d4-4d6f-9f8b-d28695a716c0
```

Step 2: Add the VMware vCenter using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

VMware vCenter JSON Files and Properties

Definition File: **omc_vmware_vcenter_sample.json**

- **omc_virtual_mgmt_system_id**: VMware vCenter Server Instance UUID
- **omc_virtual_type**: VMware
- **omc_dispatch_url**: vmware-https://<Fully qualified host name or IP address of vCenter>/sdk/vimservice

Credential File: **omc_vmware_vcenter_sample_creds.json**

- **username**: VMware vCenter username (use Administrator role).
 - **password**: VMware vCenter password (use Administrator role).
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file CREDENTIAL_FILE
[-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of VMware vCenter, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

Add ZFS Storage Appliance

You can add ZFS Storage Appliance entities using the Add Entity UI or using the cloud agent command line interface (`omcli`) with the appropriate JSON files.

Step 1: Decide how you want to add the ZFS Storage Appliance.

You can add ZFS Storage Appliance entities using one of two ways:

- Add them from UI
- Use the agent's `omcli add_entity` command with the appropriate JSON files

Adding Entities from the UI

1. From the Management Cloud main menu, select Administration, Discovery, and then Add Entity. The Add Entity page displays.
2. Select the ZFS Storage Appliance Entity Type.

3. Enter the following UI properties.

ZFS Storage Appliance UI Fields

- **Entity Name:** Name of this entity displayed in the Oracle Management Cloud console.
- **ZFFSA IP Address:** IP address of the ZFS storage appliance with REST API.
- **ZFFSA Port:** Port of the storage appliance REST API.
- **Trust Any Server Certificate:** False is recommended. You must import the storage server SSL certificate into the selected cloud agent before discovery. If you choose True instead, discovery will occur even for an untrusted or expired certificate.
- **Cloud Agent:** Cloud agent monitoring the host where the ZFS Storage Appliance is installed.

Monitoring Credentials (ZFFSA Credentials)

- **Username:** Storage appliance username.
 - **Password:** Storage appliance password.
-

See [Add Entities from the Console](#) for detailed instructions on using the Add Entity UI.

Using `omcli` and the Appropriate JSON Files

1. Download and extract the required JSON file(s) from the [master JSON zip file](#). See the table below for the specific JSON files you'll need.
2. Edit the file(s) and specify the requisite properties shown below.

ZFS Storage Appliance JSON Files and Properties

Definition File: `omc_oracle_zfs_storage_appliance_sample.json`

- **`omc_zfssa_hostname`:** ONLY IP of the ZFS Storage Appliance (if you use hostname/fully qualified domain name, you will trigger a REST fetchlet problem with certificate validation: (javax.net.ssl.SSLProtocolException: handshake alert: unrecognized_name))
- **`omc_zfssa_port`:** Port to use for REST API communication with ZFS Storage Appliance storage
- **`omc_ssl_trust_server_cert`:** Flag indicating whether to trust self-signed certificates.

Credential File: `omc_oracle_zfs_storage_appliance_sample_creds.json`

- **`Alias`:** Alias (username/login name) to be used for the ZFS Storage Appliance REST API
 - **`Password`:** Password for the ZFS Storage Appliance REST API alias.
-

3. Add the entity using `omcli`.

```
omcli add_entity agent DEFINITION_FILE [-credential_file
CREDENTIAL_FILE [-encryption_method_gpg]]
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent DEFINITION_FILE
```

See step 4. [Adding Entities to Your Service](#) of [Add Entities Using JSON Files](#) for more information.

Step 3: (Optional but recommended) Set up alerts.

To enable lights-out monitoring, you can set up alert rules to generate alerts and send notifications if your entities have performance issues.

See [Set Up Alert Rules](#) and [Set Up Alert Thresholds and Notifications](#).

Troubleshooting

If you run into any issues regarding discovery or monitoring of ZFS Storage Appliance, see the following:

- [Lack of Data](#)
- [Create an Agent Support Bundle](#)

D

Agent-monitored Entity Types and Cloud Services

The following table lists the JSON files associated with each entity type. For a thorough description of JSON file parameters for each entity type, see [Download and Customize Oracle Infrastructure Monitoring JSONs](#).

Table D-1 Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Linux Host (including Oracle Cloud Compute and Amazon EC2 that can be monitored as Linux Hosts)	update_host_sample_1.14_and_on.json	omc_host_linux	See Enable Host Monitoring for more information about Host Entities.
Solaris Host	update_host_sample_1.14_and_on.json	omc_host_solaris	
AIX Host	update_host_sample_1.14_and_on.json	omc_host_aix	
Windows Host	update_host_sample_1.14_and_on.json	omc_host_windows	See Enable Host Monitoring for more information about Host Entities.
MySQL Database	omc_mysql_database_sample.json omc_mysql_credentials.json	omc_mysql_db	
Oracle Database — Single Instance (including Database as a Service that can be monitored as an Oracle Database)	omc_oracle_database_sample.json omc_oracle_database_credentials.json	omc_oracle_db	When a CDB is added, all PDBs contained within the CDB are automatically discovered. An auto-refresh process runs periodically to discover any newly added PDBs to the CDB. Newly discovered PDBs will appear as new instances of type Oracle Pluggable Database.

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Oracle Database — Real Application Clusters (RAC)	omc_oracle_dbRAC_sample.json omc_oracle_dbRAC_sample_creds.json	omc_oracle_db	RAC instances are automatically discovered on a periodic basis once you add an Oracle RAC database. RAC instances are monitored using metrics associated with the Oracle Database. When a CDB is discovered, all PDBs contained within the CDB are automatically discovered. An auto-refresh process runs periodically to discover any newly added PDBs to the CDB. Newly discovered PDBs will appear as new instances of type Oracle Pluggable Database.
Oracle Automation Storage Management (ASM)	omc_oracle_asm_sample.json omc_oracle_asm_sample_creds.json	omc_oracle_asm	You discover RAC ASM targets one-by-one on each node. However, when discovering an ASM target on the first node, all nodes containing ASM instances will also be discovered.
Oracle Database Listener	omc_oracle_db_listener_sample.json omc_oracle_db_listener_creds.json	omc_oracle_db_listener	
Oracle Database Listener Cluster	omc_oracle_db_listener_cluster_credentials_sample.json omc_oracle_db_listener_cluster_sample.json omc_oracle_db_listener_cluster_sample_credentials.json	omc_oracle_db_listener_cluster	Cluster Listener discovery adds the following: <ol style="list-style-type: none"> Cluster Listener/SCAN Listener (Entity Type: omc_oracle_db_listener_cluster) All the Nodes/ SI Listener of Cluster (omc_oracle_db_listener)
Oracle HTTP Server (OHS)	omc_oracle_apache_sample.json omc_oracle_apache_sample_credentials_ohs11.json omc_oracle_apache_sample_credentials_ohs12.json		
Oracle SOA Infrastructure	omc_oracle_soainfra_sample_credentials.json omc_oracle_soainfra_sample.json	omc_oracle_soainfra	
Oracle Service Bus	omc_oracle_servicebus_sample.json omc_oracle_servicebus_sample_credentials.json	omc_oracle_servicebus	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Microsoft SQL Server Database	omc_sqlserver_db_sample.json omc_sqlserver_db_creds.json	omc_sqlserver_db	
MongoDB Database	omc_mongodb.json omc_mongodb_credentials.json	omc_mongodb	
Tomcat	omc_tomcat_security_sample.json omc_tomcat_credentials_sample.json	omc_tomcat	<p>If a Tomcat entity has been added to APM, it can be automatically discovered and monitored by Oracle Infrastructure Monitoring if the following conditions have been met:</p> <ul style="list-style-type: none"> • Tomcat configuration is credential-less. • The Cloud Agent resides on the same machine as Tomcat. • The Cloud Agent must be at least version 1.17.
WebLogic Server (including Java Cloud Service that can be monitored as a WebLogic Server)	N/A	omc_weblogic_j2eeserver	See the comment for WebLogic Domain.
WebLogic Cluster	N/A	omc_weblogic_cluster	See the comment for WebLogic Domain.

**Note:**

In order to auto-discover Tomcat Entities for Oracle Infrastructure Monitoring, make sure Tomcat entities are configured to make remote JMX connections.

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
WebLogic Domain	omc_weblogic_domain_sample.json omc_weblogic_domain_creds.json	omc_weblogic_domain	When you add a WebLogic Domain, and all of the WebLogic Clusters and WebLogic Servers in the domain are automatically added. There is no need to add servers and clusters separately.

 **Note:**

Only T3 and T3S protocols are supported for RMI communication between WebLogic Server and other Java programs.

When you add a WLS Domain entity using omcli (where the property capability=monitoring), the entity initially will not appear in the Oracle Infrastructure Monitoring UI.

To get the WLS Domain entity to display, navigate to the license UI, select the entity and change the edition on the WLS Domain. This will also change the edition for all members of the domain. Once done, the WLS Domain entity and any members will appear in the Oracle Infrastructure Monitoring UI.

Apache HTTP Server	omc_generic_apache_sample.json omc_oracle_apache_sample.json	omc_generic_apache	
--------------------	---	--------------------	--

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Docker Engine/ Container	omc_docker_engi ne_sample.json (used without the omc_docker_engi ne_sample_creds. json) omc_docker_engi ne_secure_sampl e.json (used with the omc_docker_engi ne_sample_creds. json) omc_docker_engi ne_sample_creds. json	omc_docker_en gine	You add docker containers by adding the Docker Engine that manages the containers. Once a Docker Engine has been added, any managed containers are automatically added. An auto-refresh process runs periodically to discover new containers.
Traffic Director Instance	omc_oracle_otd_c luster_sample.json omc_oracle_otd_c luster_sample_cre ds.json	omc_oracle_otd _instance	The Oracle Traffic Director JSON files listed here can only be used with OTD 11g. Beginning with OTD 12, Oracle Traffic Director Instance discovery is part of the WLS Domain discovery.
Oracle Traffic Director Cluster	omc_oracle_otd_c luster_sample.json omc_oracle_otd_c luster_sample_cre ds.json	omc_oracle_otd _cluster	
Cisco Ethernet (Catalyst) Switch	omc_cisco_eth_s witch_sample.json omc_cisco_eth_s witch_snmpv1_sa mple_creds.json omc_cisco_eth_s witch_snmpv3_sa mple_creds.json	omc_cisco_eth_ switch	
Oracle NoSQL	omc_nosql_db_sa mple.json	omc_oracle_no sql_db	
Arista Ethernet Switch	omc_arista_eth_s witch_sample.json omc_arista_eth_s witch_snmpv2_sa mple_creds.json omc_arista_eth_s witch_snmpv3_sa mple_creds.json	omc_arista_eth _switch	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Cisco Nexus Ethernet Switch	omc_cisco_nexus_eth_switch_sample.json omc_cisco_nexus_eth_switch_snmpv2_sample_credentials.json omc_cisco_nexus_eth_switch_snmpv3_sample_credentials.json	omc_cisco_nexus_eth_switch	
Juniper Ethernet Switch	omc_juniper_eth_switch_sample.json omc_juniper_eth_switch_snmpv2_credentials.json omc_juniper_eth_switch_snmpv3_sample_credentials.json	omc_juniper_eth_switch	
Xen Virtual Server/ Xen Virtual Platform	omc_xen_virtual_platform_sample.json omc_xen_virtual_platform_with_root_credentials.json omc_xen_virtual_platform_with_sudo_credentials.json omc_xen_virtual_platform_with_ssh_keys.json omc_xen_virtual_platform_with_sudo_ssh_keys.json	omc_xen_virtual_platform	
Infiniband Switch	omc_oracle_ib_switch_sample.json omc_oracle_ib_switch_snmpv2_sample_credentials.json omc_oracle_ib_switch_snmpv3_sample_credentials.json	omc_oracle_ib_switch	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Power Distribution Unit (PDU)	omc_oracle_pdu_sample.json omc_oracle_pdu_sample_snmpv1_creds.json omc_oracle_pdu_sample_snmpv3_creds.json	omc_pdu	
Oracle Fabric Manager / Virtual Networking / Xsigo	omc_oracle_ovn_sample_creds.json omc_oracle_ovn_sample.json	omc_oracle_ovn	
Brocade Fibre Channel Switch	omc_brocade_fc_switch_sample.json omc_brocade_fc_switch_snmpv1_sample_creds.json omc_brocade_fc_switch_snmpv3_sample_creds.json	omc_brocade_fc_switch	
Oracle HTTP Server (OHS)	omc_oracle_apache_sample.json omc_oracle_apache_sample_creds_ohs11.json omc_oracle_apache_sample_creds_ohs12.json	omc_oracle_apache	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
SCOM (System Center Operations Manager)	omc_microsoft_sc om_sample.json omc_microsoft_sc om_sample_creds .json	omc_microsoft_ scom	<p>This integration can be used to monitor Windows host servers by retrieving the host performance data from Microsoft SCOM.</p> <p>SCOM Integration (Windows host)</p> <ol style="list-style-type: none"> 1. Deploy a Cloud Agent on the Windows machine. 2. Copy the following files (available on the SCOM server) required by the Cloud agent: to this location: <pre>%ProgramFiles%\Microsoft System Center 2012 R2\Operations Manager\Server\SDK Binaries\Microsoft.Enterp riseManagement.Runtime.dl l</pre> <pre>%ProgramFiles%\Microsoft System Center 2012 R2\Operations Manager\Server\SDK Binaries\Microsoft.Enterp riseManagement.Operations Manager.dll</pre> <pre>%ProgramFiles%\Microsoft System Center 2012 R2\Operations Manager\Server\SDK Binaries\Microsoft.Enterp riseManagement.Core.dll</pre>

 **Note:**

If the Cloud agent is deployed on the same Windows server as SCOM, there is no need to copy these files.

Table D-1 (Cont.) Agent-monitored Entity Types


Entity Type	JSON Files	Entity Internal Name	Comments
			<p>3. Uncomment the following line in <i>discovery.properties</i> file:</p> <pre>disable_monitoring_for_entitytype=omc_host_windows</pre> <p>The Windows host has already been added to OMC by the Cloud Agent., meaning this SCOM integration uses the windows host that is already added by the Cloud Agent and then starts populating its metrics. Hence, there is no need to have add duplicate Wndows host entities.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> Note:</p> <p>There is no need to bounce the Cloud Agent.</p> </div> <p>Once successfully integrated, OMC automatically refreshes every minute to detect new host Windows servers.</p> <p>Monitoring of the following entities is supported:</p> <ul style="list-style-type: none"> • Windows Hosts • SQL Server DB • Exchange Server • Hyper-V (via Virtual Platform and Virtual Server) • Active Directory • IIS
Juniper SRX Firewall	omc_juniper_srx_sample.json omc_juniper_srx_snmpv2_sample_creds.json omc_juniper_srx_snmpv3_sample_creds.json	omc_juniper_srx	
Fujitsu Computers	omc_fujitsu_server_sample.json omc_fujitsu_server_creds_sample.json	omc_fujitsu_server	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Intel/SPARC Computers	omc_ilom_server.json omc_ilom_server_creds.json	omc_ilom_server	
VMware vCenter	omc_vmware_vcenter_sample.json omc_vmware_vcenter_sample_creds.json	NA	vCenter is a composite entity consisting of the following entities: <ul style="list-style-type: none"> • Virtual Management System • Datacenter • Datastore • Resource Pool • Virtual cluster • Virtual application • Virtual Platform • Virtual Server
NGINX	omc_nginx.json	omc_nginx	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Docker Swarm	<p>Entity JSONs for Docker Swarm: <i>Adding Non-Secure Docker Swarm Target</i> Add_Entity_Docker_Swarm_Non_Secure.json <i>Adding 1WAY Docker Swarm Target</i> Add_Entity_Docker_Swarm_1way_SSL.json <i>Adding 2WAY Docker Swarm Target</i> Add_Entity_Docker_Swarm_2way_SSL.json</p> <p>Credential JSONs: :Docker_Swarm_Secure_Credentials.json</p> <p>Entity JSONs for Docker Worker Engines: <i>Adding Non-Secure Docker Worker Engine</i> Add_Entity_Worker_Docker_Engine_Non_Secure.json <i>Adding 1WAY Docker Worker Engine</i> Add_Entity_Worker_Docker_Engine_1way_SSL.json <i>Adding 2WAY Docker Worker Engine</i> Add_Entity_Worker_Docker_Engine_2way_SSL.json</p>	omc_docker_swarm	<p>The Docker Engine credential json remains the same for Worker Engines.</p> <p>For secure mode, apart from the jsons, you need to add the docker truststore certificate(CA certificate) in the agent default truststore(\$EMSTATE/sysman/config/montrust/AgentTrust.jks).</p> <p>Command: omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc <certificate location> -alias dockercertificate</p> <p>For example: slce03.cer is the CA certificate. omcli secure add_trust_cert_to_jks -password welcome -trust_certs_loc /certificate_directory/slce03.cert -alias dockercertificate</p> <p>Fetching Swarm ID: Do a GET on LEADER_BASE_URL/swarm For example: GET on http://myserver.mycompany.com:4243/swarm</p>

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Apache SOLR	Entity JSONs omc_solr_instance_credless.json omc_solr_instance_creds.json omc_solrcloud_credless.json omc_solrcloud_creds.json Credential JSONs solr_basic_authentication.json solr_client_authentication.json solr_client_with_basic_authentication.json	omc_solr_instance	
Arbor Networks TMS	omc_arbor_tms_sample.json omc_arbor_tms_snmpv2_sample_creds.json omc_arbor_tms_snmpv3_sample_creds.json	omc_arbor_tms	
Arbor Networks CP	omc_arbor_cp_sample.json omc_arbor_cp_snmpv2_sample_creds.json omc_arbor_cp_snmpv3_sample_creds.json	omc_arbor_cp	
Juniper Netscreen Firewall	omc_juniper_netscreen_sample.json omc_juniper_netscreen_snmpv2_sample_creds.json omc_juniper_netscreen_snmpv3_sample_creds.json	omc_juniper_netscreen	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Juniper MX Router	omc_juniper_mx_sample.json omc_juniper_mx_snmpv2_sample_credentials.json omc_juniper_mx_snmpv3_sample_credentials.json	omc_juniper_mx	
F5 BIG-IP LTM	omc_f5_bigip_ltm_sample.json omc_f5_bigip_ltm_snmpv2_sample_credentials.json omc_f5_bigip_ltm_snmpv3_sample_credentials.json	omc_f5_bigip_ltm	
F5 BIG-IP DNS	omc_f5_bigip_dns_sample.json omc_f5_bigip_dns_snmpv2_sample_credentials.json omc_f5_bigip_dns_snmpv3_sample_credentials.json	omc_f5_bigip_dns	
Hadoop Cluster	No Credentials hadoop_credless.json Credentials hadoop_credentials.json Credential Input File hadoop_credentials_input.json	omc_oracle_hadoop_cluster omc_oracle_hadoop_hdfs omc_oracle_hadoop_yarn omc_hadoop_datanode omc_hadoop_namenode omc_hadoop_nodemanager omc_hadoop_resourcemanager	When a Hadoop Cluster is added, the following entities of the Hadoop environment are automatically discovered. . <ul style="list-style-type: none"> • Hadoop HDFS • Hadoop YARN • Hadoop Namenode • Hadoop Datanode • Hadoop Nodemanager • Hadoop Resourcemanager
NetApp FAS	omc_netapp_fas_sample.json omc_netapp_fas_snmp_sample_credentials.json	omc_netapp_fas	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
ZFS Storage Appliance	omc_oracle_zfs_storage_appliance_sample.json omc_oracle_zfs_storage_appliance_sample_creds.json	omc_oracle_zfs_storage_appliance	
Kubernetes	Entity JSONs (Without Heapster & Heapter running as cluster service) omc_kubernetes_cluster_insecure.json omc_kubernetes_cluster_secure.json Entity JSONs (With Heapster not running as cluster service) omc_kubernetes_cluster_insecure(heapster).json omc_kubernetes_cluster_secure(heapster).json Credential JSONs alias_creds.json keystore_creds.json token_creds.json	omc_kubernetes	
ES2 Ethernet Switch	omc_es2_sample.json omc_es2_snmpv2_sample_creds.json omc_es2_snmpv3_sample_creds.json	omc_es2	
Oracle Flash Storage	omc_oracle_flash_storage_sample.json omc_oracle_flash_storage_creds_sample.json	omc_oracle_flash_storage	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Cassandra DB	omc_cassandra_db.json omc_cassandra_db_creds.json	omc_cassandra_db	
EMC VMAX	omc_emc_vmax_sample.json omc_emc_vmax_creds_sample.json	omc_emc_vmax	
EMC VNX	omc_emc_vnx_creds_sample.json omc_emc_vnx_instance_sample.json	omc_emc_vnx	
L2/L3 Generic Network Node	omc_network_node_sample.json omc_network_node_snmpv2_sample_creds.json omc_network_node_snmpv3_sample_creds.json	omc_network_node	
Oracle VM Server for SPARC (LDOM)	omc_sparc_ldoms_sample.json	omc_sparc_ldoms	
JBoss	omc_jboss_domain_sample.json omc_jboss_domain_sample_creds.json omc_jboss_domain_secure_sample.json omc_jboss_domain_secure_sample_creds.json omc_jboss_standalone_j2eeserver_sample.json omc_jboss_standalone_j2eeserver_sample_creds.json omc_jboss_standalone_j2eeserver_secure_sample.json omc_jboss_standalone_j2eeserver_secure_sample_creds.json	omc_jboss_domain omc_jboss_standalone	

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Oracle Coherence	omc_oracle_coherence.json omc_oracle_coherence_cred.json coherence_credentials.json	omc_oracle_coherence	
Oracle Clusterware (CRS)	omc_oracle_clusterware_credless_sample.json omc_oracle_clusterware_sample.json omc_oracle_clusterware_credential_sample.json	omc_oracle_clusterware	<p>CRS discovery adds:</p> <ol style="list-style-type: none"> 1. CRS (Entity Type: omc_oracle_clusterware) 2. All the cluster nodes (omc_cluster_node) of the CRS. The cluster nodes are discovered automatically when CRS is added. <p>Both local and remote monitoring is supported.</p> <p>Cluster node and CRS have the following association :</p> <p><i>omc_oracle_clusterware</i> contains <i>omc_cluster_node</i></p>
Oracle GoldenGate	omc_oracle_goldengate_sample_arc_h_classic.json omc_oracle_goldengate_sample_arc_h_microservice.json omc_oracle_goldengate_sample_creds.json	omc_oracle_goldengate	
Oracle VM Manager	omc_oracle_vm_manager.json omc_oracle_vm_manager_creds.json	omc_oracle_vm_manager omc_oracle_vm_zone omc_oracle_vm_server_pool omc_ovm_virtual_platform omc_ovm_virtual_server	<p>Oracle VM Manager is a composite entity consisting of the following entities:</p> <ul style="list-style-type: none"> • Oracle VM Manager • Oracle VM Zone • Oracle VM Server Pool • Oracle VM Virtual Platform • Oracle VM Virtual Server <p>The user provides a json file with details to discover Oracle VM Manager, then all the Oracle VM Manager related entities such as Oracle VM Zone, Oracle VM Server Pool , Oracle VM Virtual Platform and Oracle VM Virtual Server are automatically discovered and updated periodically when things change in the Oracle VM Manager deployment.</p>

Table D-1 (Cont.) Agent-monitored Entity Types

Entity Type	JSON Files	Entity Internal Name	Comments
Oracle JVM	omc_jvm_sample.json omc_jvm_sample_creds	omc_jvm	

Table D-2 Supported Cloud Services

Cloud Vendor	Cloud Service	Comments
Oracle Cloud	Compute	Can be also monitored via Cloud agent for more comprehensive monitoring.
Amazon AWS	Elastic Cloud Compute (EC2)	Can be monitored via Cloud agent for more comprehensive monitoring.
Amazon AWS	Relational Database Service (RDS)	For RDS (Oracle) can be monitored via Cloud agent for more comprehensive monitoring.
Amazon AWS	Simple Storage Service (S3)	
Amazon AWS	Elastic Block Store (EBS)	
Amazon AWS	Lambda	
Amazon AWS	Redshift	
Amazon AWS	Elastic Load Balancer (ELB)	
Amazon AWS	Elastic Load Balancer (ELB) - Application Load Balancer	
Amazon AWS	Simple Queue Service (SQS)	
Amazon AWS	Simple Notification Service (SNS)	

E

Monitor AWS - RDS Oracle DB

1. Discover an AWS RDS DB instance in Oracle Management Cloud.
 - a. Request an EC2 Instance (Amazon Linux AMI or Red Hat Enterprise Linux)
 - b. Request an RDS - Oracle DB creation
 - c. Install SQLclient on the EC2 instance and connect to the RDS - Oracle DB instance and grant the permissions are listed below.
 - d. Install the Cloud agent on the EC2 instance.
 - e. Discover the RDS - Oracle DB using the Cloud agent to Oracle Management Cloud.
2. Grant the requisite privileges.

Create the monitoring user and give the required grants shown below in **Setting up Infrastructure Monitoring of an Amazon Oracle RDS Instance**.

In addition, you also need to grant the following privileges:

```
exec
rdsadmin.rdsadmin_util.grant_sys_object('CDB_SERVICES','MONCSROLE','SELECT
');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SERVICES','MONCSROLE','SELECT
');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$SERVICES','MONCSROLE','SELECT
');
exec
rdsadmin.rdsadmin_util.grant_sys_object('CDB_PDBS','MONCSROLE','SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$CONTAINERS','MONCSROLE','SELE
CT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$CONTAINERS','MONCSROLE','SELE
CT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('CDB_TABLESPACES','MONCSROLE','SEL
ECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('CDB_DATA_FILES','MONCSROLE','SELE
CT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('CDB_TEMP_FILES','MONCSROLE','SELE
CT');
```

Setting up Infrastructure Monitoring of an Amazon Oracle RDS Instance

1. Create "moncsrole" and "moncs" user in the AWS Oracle RDS Instance..

```
SQL> create role moncsrole;
Role created.
SQL>
SQL> create user moncs identified by <password>;
User created.
SQL>
SQL> grant moncsrole to moncs;
Grant succeeded.
SQL>
SQL> grant create session to moncs;
Grant succeeded.
SQL>
```

2. Grant the required privileges to "moncsrole" created above. Some of the grants are executed differently than those for a regular Oracle Instance. This is documented in the AWS RDS documentation.

```
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$PARAMETER','MONCSROLE',
'SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$INSTANCE','MONCSROLE','S
ELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SQL','MONCSROLE','SELEC
T');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$DATABASE','MONCSROLE','S
ELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$INSTANCE','MONCSROLE','
SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$OSSTAT','MONCSROLE','SE
LECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SGA','MONCSROLE','SELEC
T');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$PGASTAT','MONCSROLE','S
ELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SYSMETRIC_SUMMARY','MON
CSROLE','SELECT');
grant select on sys.dba_tablespaces to moncsrole;
grant select on dba_data_files to moncsrole;
grant select on dba_free_space to moncsrole;
exec
rdsadmin.rdsadmin_util.grant_sys_object('DBA_UNDO_EXTENTS','MONCSROL
E','SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$ACTIVE_SESSION_HISTORY',
```

```

'MONCSROLE','SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$ASH_INFO','MONCSROLE','SELECT
');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$PARAMETER','MONCSROLE','SELECT
');
grant select on dba_temp_files to moncsrole;
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SORT_SEGMENT','MONCSROLE','SE
LECT');
grant select on sys.ts$ to moncsrole;
grant execute on sys.dbms_lock to moncsrole;
grant execute on dbms_system to moncsrole;
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$IOSTAT_FILE','MONCSROLE','SEL
ECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SYSSTAT','MONCSROLE','SELECT
');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SYS_TIME_MODEL','MONCSROLE','
SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$EVENT_NAME','MONCSROLE','SELE
CT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$WAITCLASSMETRIC','MONCSROLE',
'SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SYSTEMMETRIC','MONCSROLE','SELE
CT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SYSTEM_EVENT','MONCSROLE','SE
LECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('GV_$SQL','MONCSROLE','SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$SYSTEM_EVENT','MONCSROLE','SEL
ECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$ALERT_TYPES','MONCSROLE','SELE
CT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$THRESHOLD_TYPES','MONCSROLE','
SELECT');
exec
rdsadmin.rdsadmin_util.grant_sys_object('V_$SYSTEMMETRIC','MONCSROLE','SELE
CT
');
```

3. Add the Oracle DB entity.

```

omcli add_entity agent omc_oracle_db_oral2c.json -credential_file
omc_oracle_db_oral2c_creds.json
```

4. Verify the status of the newly added entity.

```
omcli status_entity agent omc_oracle_db_oral2c.json
```

F

Configure a Coherence Cluster

The following section covers the configuration procedure for a standalone Coherence cluster. For details on configuring a managed Coherence cluster, refer to the WebLogic documentation.

Overview

Oracle Coherence standalone deployments can be monitored using Oracle Management Cloud by configuring the Coherence nodes with a set of Coherence and JMX system properties (start arguments). In addition, one of the nodes will have to be configured as a central JMX management node. This JMX management node must expose all Coherence MBeans and attributes. See [Creating and Starting a JMX Management Node](#) for details. In addition to configuring the JMX management node, the Cloud Agent must also be installed and configured on the same host as JMX management node. This is required to discover and monitor the Coherence cluster in Oracle Management Cloud.

Coherence Management (JMX) node's MBean server will expose MBeans for entire Coherence cluster. Enterprise Manager will connect to this management node to discover and monitor Coherence cluster.

Creating and Starting a JMX Management Node

The Cloud Agent uses the JMX management node (centralized MBean server) to discover and monitor the entire Coherence cluster, including the nodes and caches. As a best practice, it is recommended that the Management Agent be present on the same host as the JMX management node that is used to discover and monitor the Coherence cluster. The Management Agent must be setup on all the machines on which the Coherence nodes are running to monitor and provision the cluster. To configure the JMX management node, you must:

- Specify Additional System Properties
- Include Additional Class Path
- Use the Enterprise Manager Custom Start Class

Specifying Additional System Properties

Note:

Oracle recommends that the management node is configured as a storage disabled node to ensure minimal performance impact on any Coherence caches.

The following start arguments must be added to one of the Coherence nodes to configure it as the JMX central management node.

- `-Dtangosol.coherence.management.extendedmbeanname=true` (allows any restarted node to be automatically detected by Enterprise Manager. This parameter is available in Coherence 3.7.1.9 and later versions)

- If set to true, the status of the node is automatically refreshed when a node is restarted.
- If this property is not set, you must use the Refresh Cluster option to update the status of a node when it is restarted.
- If you start a node after setting this property to true, all nodes in the cluster must be started after the `extendedmbeanname` property is set to true.
- `-Dtangosol.coherence.management=all` (enables monitoring for all nodes)
- `-Dcom.sun.management.jmxremote.port=<port number>` (required for remote connection for coherence 12.2.1.x or older versions.)
- `-Dtangosol.coherence.distributed.localstorage=false` (disables caching and ensures that the node is a dedicated monitoring node)
- `-Doracle.coherence.home=<coherence home>`
- `-Dtangosol.coherence.member=<member name>` (required for target name)
- `-Doracle.coherence.machine=<fully qualified hostname>` (must match the name of the host discovered in Enterprise Manager)

 **Note:**

If you are using JMX credentials, you must set the following additional start arguments.

- `-Dcom.sun.management.jmxremote.ssl=true`
- `-Dcom.sun.management.jmxremote.authenticate=true`

If no JMX credentials are used, you must set these arguments to **false**.

Including the Additional Class Path

You must include the path to the Enterprise Manager custom jar files, `coherenceEMIntg.jar` and the `bulkoperationsmbean.jar` for clusters versions older than 12.2.1. These jar files are available in the following locations:

```
<AGENT_BASE_DIRECTORY>/plugins/oracle.em.sgfm.zip/<VERSION 1.26 or above>/
archives/bulkoperationsmbean.jar
```

```
<AGENT_BASE_DIRECTORY>/plugins/oracle.em.sgfm.zip/<VERSION 1.26 or above>/
archives/coherenceEMIntg.jar
```

Coherence cluster with version 12.2.1 and above must use the `coherenceEMIntg.jar` file available in the

```
<AGENT_BASE_DIRECTORY>/plugins/oracle.em.sgfm.zip/<VERSION 1.26 or above>/
archives/12.2.1 directory.
```

 **Note:**

The location of the .jar files may change based on the plugin version.

Using the Custom Start Class

In addition to configuring the system properties and the class path when starting Coherence management node, it is also required that you use the Enterprise Manager EMIntegrationServer class as the start class. This class allows you to register the custom MBeans required for the Cache Data Management feature of Management Pack for Oracle Coherence.

Example Start Script for the Coherence Management Node

An example start script for the management node is given below:

```
#
#!/bin/sh

CP=$CP:<EM_CC_Agent_Home>/plugins/oracle.sysman.emas.agent.plugin_12.1.0.6.0/
archives/coherence/coherenceEMIntg.jar:
<EM_CC_Agent_Home>/plugins/oracle.sysman.emas.agent.plugin_12.1.0.6.0/
archives/coherence/bulkoperationsmbean.jar
COH_OPTS="$COH_OPTS -cp $CP"
$JAVA_HOME/bin/java $COH_OPTS
-Dtangosol.coherence.management.extendedmbeaname=true
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.ssl=false
-Dtangosol.coherence.management=all
-Dtangosol.coherence.member=<unique member name>
-Doracle.coherence.machine=<hostname_as_discovered_in_EM>
-Dcom.sun.management.jmxremote.port=<OpenTCP_Port>
-Doracle.coherence.home=$COHERENCE_HOME
-Dtangosol.coherence.distributed.localstorage=false
-Dtangosol.coherence.management.refresh.expiry=1m
-server
-Xms2048m -Xmx2048m
oracle.sysman.integration.coherence.EMIntegrationServer
```

Configuring All Other Nodes

In addition to configuring the Coherence JMX management node, you must configure all other Coherence cluster nodes with additional Coherence specific system properties (start arguments) used by Oracle Management Cloud.

Additional System Properties for All Other Coherence Nodes

The following system properties must be added to all other Coherence nodes.

```
-Dtangosol.coherence.management.extendedmbeaname=true
-Dtangosol.coherence.management.remote=true -
Dtangosol.coherence.member=<unique member name> -
Doracle.coherence.home=<coherence home>
-Doracle.coherence.machine=<machine name> should be the same as the name of
the host discovered in Oracle Management Cloud.
```

 **Note:**

If you are using JMX credentials, you must set the following additional start arguments.

- `-Dcom.sun.management.jmxremote.ssl=true`
- `-Dcom.sun.management.jmxremote.authenticate=true`

If no JMX credentials are used, you must set these arguments to **false**.

Example Start Script for All Other Coherence Nodes

An example start script for all other Coherence nodes is given below:

```
#!/bin/sh

COH_OPTS="$COH_OPTS -cp $CP"
$JAVA_HOME/bin/java $COH_OPTS
-Dtangosol.coherence.management.extendedmbeanname=true
-Dtangosol.coherence.management.remote=true
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-Doracle.coherence.home=<coherence home>
-Dtangosol.coherence.member=<unique member name>
-Doracle.coherence.machine=<hostname>
-Dcom.tangosol.net.DefaultCacheServer
```

Testing the Configuration

To test the Coherence cluster configuration for use in Oracle Management Cloud, you must verify that the central management (JMX) node has information regarding the managed objects of all other Coherence cluster nodes, caches, services, and so on. Additionally, you must verify that the central management node is accessible remotely, either through `<hostname>:<port>` OR the JMX Service URL. If JMX credentials are used, they should also be specified.

Verifying Remote Access for the MBean Objects Using JConsole

JConsole is a Java tool available through JDK. You can use this to verify remote access to the MBean objects of entire Coherence cluster nodes, caches, services, and so on.

To verify remote access, open JConsole and select "New Connection". In New Connection page, select **Remote Process** and provide connection details where `<hostname>` is the name of the machine where central management node is running, `<port>` is what you have specified in the -

`Dcom.sun.management.jmxremote.port` parameter while starting the management node. If successful, you will see the MBean object tree.

If you see MBeans for all Coherence nodes in the System MBean Browser or JConsole, you can now discover and monitor the Coherence cluster and its associated elements in Oracle Management Cloud.

G

Additional collectd Configurations and Information

This appendix contains the following:

- [Manual Mapping](#)
- [Metric Schema Mapping \(collectd\)](#)
- [Send a Subset of collectd Metrics to Oracle Management Cloud](#)
- [Receive Metrics from a Remote Generic Metric Collector](#)
- [Troubleshooting collectd Metric Collection](#)

Manual Mapping

As an alternative to automatic data mapping, you can manually map metric data to handle monitoring requirements not covered with automatic mapping.

Manual mapping can provide greater flexibility in mapping data when automatic mapping doesn't fit your monitoring requirements.



Note:

The metric schema automatically defined with automatic mapping is described in [Metric Schema Mapping \(collectd\)](#).

The metric schema automatically defined with automatic mapping is described in [Metric Schema Mapping \(collectd\)](#). If you feel this definition does not suit your requirement, you can try manually mapping the metrics. For this you'll need to perform the following steps:

1. Define your own destination metric in Oracle Management Cloud using REST APIs.
[Example: Generic Metric Collector Entity Type \(Auto-mapping\)](#)
2. Write a mapping metadata JSON file to map source collectd metrics to the destination metrics.
[Example: Mapping Metadata](#)
3. Provide a path to the mapping metadata JSON file when adding the Generic Metric Collector entity.
[Example: Destination Metric Definitions](#)

Example: Generic Metric Collector Entity (Manual Mapping)

The following example shows a generic metric collector entity named *collectd-myhost* that is locally monitored on the Cloud agent host *myhost*.

Example G-1 Entity of Generic Metric Collector Type - Locally Monitored

```
{
  "entities":
  [
    {
      "name": "collectd-myhost",
      "type": "omc_generic_metric_collector",
      "displayName": "collectd-myhost",
      "timezoneRegion": "PDT",
      "properties":
      {
        "host_name":
        {
          "displayName": "Host Name",
          "value": "myhost.mycompany.com"
        },

        "capability":
        {
          "displayName": "capability",
          "value": "monitoring"
        },

        "omc_monitored":
        {
          "displayName": "Cloud Agent Monitored",
          "value": "TRUE"
        },

        "omc_query_interface_path":
        {
          "displayName": "Query Interface Path",
          "value": "/opt/collectd/bin/collectdctl"
        },

        "omc_product_name":
        {
          "displayName": "Product Name",
          "value": "collectd"
        },

        "omc_product_vendor":
        {
          "displayName": "Product Vendor",
          "value": "Florian octo Forster, et al."
        },

        "omc_product_version_query_arg":
        {
          "displayName": "Product Version Query Argument",
          "value": "-h"
        },

        "omc_product_version_regex":
```

```
{
  "displayName": "Product Version Regular Expression",
  "value": "^collectd (.+), http"
},

"omc_metrics_query_arg":
{
  "displayName": "Metrics Query Argument",
  "value": "listval"
},

"omc_response_query_arg":
{
  "displayName": "Response Query Argument",
  "value": "listval"
},

"omc_use_exit_code_for_response":
{
  "displayName": "Use exit code for response",
  "value": "TRUE"
},

"omc_protocol":
{
  "displayName": "Protocol",
  "value": "https"
},

"omc_payload_format":
{
  "displayName": "Payload Format",
  "value": "json"
},

"omc_filter_expression":
{
  "displayName": "Filter Expression",
  "value": "{$.[?(@.host=='myhost.mycompany.com')]}"}
},

"omc_mapping_metadata_file_path":
{
  "displayName": "Mapping Metadata File Path",
  "value": "/scratch2/agent/gmc/mapping_metadata_processes.json"
}
}
]
}
```

Example: Mapping Metadata

The following example illustrates the how to map collectd metrics to Oracle Management Cloud metrics.

Example G-2 Mapping Metadata - Processes

```
{
  "entityMetricMappings":
  [
    {
      "entityTypeSourceFilter":
      {
        "value": "{$[?(@.plugin=='processes')]}"
      },

      "entityType": "%host_type%",
      "entityName":
      {
        "value": "%$host%"
      },

      "metricGroupMappings":
      [
        {
          "metricGroupSourceFilter":
          {
            "in":
            {
              "field": "type",
              "values":
              [
                "fork_rate",
                "ps_state"
              ]
            }
          }
        },

        "metricGroupName":
        {
          "join":
          {
            "values":
            [
              "collectd_processes",
              "%$type%",
              "%$type_instance%"
            ],

            "delimiter": "_"
          }
        },

        "metricNames":
```

```

    [
      {
        "join":
        {
          "values":
          [
            "processes",
            "%$type%",
            "%$type_instance%"
          ],
          "delimiter": "_"
        }
      }
    ],
    "metricValues":
    [
      {
        "value": "%$values[0]%"
      }
    ]
  }
},
"metricCollectionTimeMapping":
{
  "value": "%$time%",
  "timeFormat": "UNIX"
}
}

```

Example: Destination Metric Definitions

The following example destination metric definition is posted to the entity model REST API

Path: /entityModel/metadata/entityTypes/omc_host/metricGroupTypes

Example G-3 Destination Metric Definitions - Custom Metric Groups on the Host Interface

```

[
  {
    "entityTypeName": "omc_host",
    "metricGroupName": "collectd_processes_ps_state_running",
    "metricGroupDisplayName": "Running processes",
    "description": "Number of processes in running state",
    "config": false,
    "parentMGName": null,
    "curationLevel": 1,
    "columnList":
    [
      {

```

```
        "metricColumnDisplayName": "Running processes",
        "metricColumnName": "processes_ps_state_running",
        "metricColumnClass": "NUM",
        "baselineable": true,
        "typeFormat": null,
        "isKey": false,
        "unitType": "OMC_SYS_STANDARD_GENERAL_NA",
        "description": "Number of processes in running state",
        "category": "Load"
    }
]
},
{
    "entityTypeName": "omc_host",
    "metricGroupName": "collectd_processes_ps_state_sleeping",
    "metricGroupDisplayName": "Sleeping processes",
    "description": "Number of processes in sleeping state",
    "config": false,
    "parentMGName": null,
    "curationLevel": 1,
    "columnList":
    [
        {
            "metricColumnDisplayName": "Sleeping processes",
            "metricColumnName": "processes_ps_state_sleeping",
            "metricColumnClass": "NUM",
            "typeFormat": null,
            "isKey": false,
            "unitType": "OMC_SYS_STANDARD_GENERAL_NA",
            "description": "Number of processes in sleeping state",
            "category": "Load"
        }
    ]
},
{
    "entityTypeName": "omc_host",
    "metricGroupName": "collectd_processes_ps_state_blocked",
    "metricGroupDisplayName": "Blocked processes",
    "description": "Number of processes in blocked state",
    "config": false,
    "curationLevel": 1,
    "columnList":
    [
        {
            "metricColumnDisplayName": "Blocked processes",
            "metricColumnName": "processes_ps_state_blocked",
            "metricColumnClass": "NUM",
            "typeFormat": null,
            "isKey": false,
            "unitType": "OMC_SYS_STANDARD_GENERAL_NA",
            "description": "Number of processes in blocked state",
            "category": "Load"
        }
    ]
}
```

```
]
},
{
  "entityTypeName": "omc_host",
  "metricGroupName": "collectd_processes_ps_state_stopped",
  "metricGroupDisplayName": "Stopped processes",
  "description": "Number of processes in stopped state",
  "config": false,
  "curationLevel": 1,
  "columnList":
  [
    {
      "metricColumnDisplayName": "Stopped processes",
      "metricColumnName": "processes_ps_state_stopped",
      "metricColumnClass": "NUM",
      "typeFormat": null,
      "isKey": false,
      "unitType": "OMC_SYS_STANDARD_GENERAL_NA",
      "description": "Number of processes in stopped state",
      "category": "Load"
    }
  ]
},
{
  "entityTypeName": "omc_host",
  "metricGroupName": "collectd_processes_ps_state_paging",
  "metricGroupDisplayName": "Paging processes",
  "description": "Number of processes in paging state",
  "config": false,
  "curationLevel": 1,
  "columnList":
  [
    {
      "metricColumnDisplayName": "Paging processes",
      "metricColumnName": "processes_ps_state_paging",
      "metricColumnClass": "NUM",
      "typeFormat": null,
      "isKey": false,
      "unitType": "OMC_SYS_STANDARD_GENERAL_NA",
      "description": "Number of processes in paging state",
      "category": "Load"
    }
  ]
},
{
  "entityTypeName": "omc_host",
  "metricGroupName": "collectd_processes_ps_state_zombies",
  "metricGroupDisplayName": "Zombie processes",
  "description": "Number of processes in zombie state",
  "config": false,
  "curationLevel": 1,
  "columnList":
```

```
[
  {
    "metricColumnName": "processes_ps_state_zombies",
    "metricColumnClass": "NUM",
    "typeFormat": null,
    "isKey": false,
    "unitType": "OMC_SYS_STANDARD_GENERAL_NA",
    "description": "Number of processes in zombie state",
    "category": "Load"
  }
],
{
  "entityTypeName": "omc_host",
  "metricGroupName": "collectd_processes_fork_rate",
  "metricGroupDisplayName": "Fork Rate",
  "description": "Overall rate of creation of processes or threads
by all CPUs",
  "config": false,
  "parentMGName": null,
  "curationLevel": 1,
  "columnList":
  [
    {
      "metricColumnName": "processes_fork_rate",
      "metricColumnClass": "NUM",
      "typeFormat": null,
      "isKey": false,
      "unitType": "OMC_SYS_STANDARD_RATE_SEC",
      "description": "Overall rate of creation of processes or
threads by all CPUs",
      "category": "Load"
    }
  ]
}
]
```

Metric Schema Mapping (collectd)

The following table shows the Oracle Management Cloud metric schema derived from collectd metric identifiers. If you enable *automatic mapping*, collectd metrics are automatically mapped to Oracle Management Cloud metrics.

Note:

The maximum number of metric groups per entity type (auto-created by Oracle Management Cloud when using the automatic mapping) is 70. On the collectd side, this translates to 70 distinct type + type_instance combinations of metric identifiers per collectd plugin.

Table G-1 Metric Schema Mapping

Field	Value	Example
parentTargetType(64)	omc_target	omc_target
entityType(64)	'_gmc_collectd_' <plugin>	_gmc_collectd_redis
typeDisplayName(128)	<plugin in Title Case, replace underscore with space>	Redis
entityName(256)	<host>'-'<plugin_instance> or <host> (if no plugin_instance)	myhost.myco.com-6379
metricGroupName(64)	<type>'_'<type_instance> or <type> (if no type_instance)	1. memcached_connections_clients 2. df_memory 3. uptime (no type_instance)
metricGroupDisplayName(256)	<metricGroupName in Title Case with underscores replaced by spaces>	Memcached Connections Clients
metricColumnName(64)	<dsname>	1. value 2. used, free 3. value
metricColumnDisplayName(256)	If <dsname> equals 'value' same as <metricGroupDisplayName> Else <metricGroupDisplayName>'<dsname in Title Case>	1. Memcached Connections Clients ('value' omitted) 2. Df Memory Used, Df Memory Free 3. Uptime (no type_instance)
metricColumnClass	NUM	NUM

You can extract the plugin name, plugin_instance, type, and type instance by running the `collectdctl listval` command.

Example:

```
$ /opt/collectd/bin/collectdctl listval
myhost.mycompany.com/load/load
myhost.mycompany.com/processes/fork_rate
myhost.mycompany.com/processes/ps_state-blocked
myhost.mycompany.com/processes/ps_state-paging
myhost.mycompany.com/processes/ps_state-running
myhost.mycompany.com/processes/ps_state-sleeping
myhost.mycompany.com/processes/ps_state-stopped
myhost.mycompany.com/processes/ps_state-zombies
```

You can obtain the dsname (data source name) by running the `collectdctl getval` command..

Example:

```
$ /opt/collectd/bin/collectdctl getval myhost.mycompany.com/load/load
shortterm=1.100000e-01
midterm=8.000000e-02
longterm=6.000000e-02
```

Send a Subset of collectd Metrics to Oracle Management Cloud

To minimize noise and increase efficiency of what is sent from collectd to Oracle Management Cloud, it is possible to write only a subset of metrics collected by collectd to Oracle Management Cloud. A *PostCacheChain* can be configured to specify this. Assume that following read plugins were initially enabled and were writing their output to the *write_log* plugin.

Initial Read Plugins Enabled

```
LoadPlugin cpu
LoadPlugin interface
LoadPlugin memory
LoadPlugin processes
```

To send the output from the *processes* plugin to Oracle Management Cloud, but not the other plugins. A *PostCacheChain* can be configured to achieve this, as shown in the following example..

Example: PostCacheChain Configuration to Selectively Send the Output of Processes to Oracle Management Cloud

```
LoadPlugin match_regex
PostCacheChain "PostCache"
<Chain "PostCache">
  <Rule "write_omc">
    <Match "regex">
      Plugin "^processes$"
    </Match>
    <Target "write">
      Plugin "write_http/omc"
      Plugin "write_log"
    </Target>
    Target "stop"
  </Rule>
  # Default target
  <Target "write">
    Plugin "write_log"
  </Target>
</Chain>
```

Here we have specified a rule called "write_omc" with a regex to match the plugin's name (processes) to be sent to Oracle Management Cloud as well as to write_log (as before). The default write target is configured not to write to Oracle Management

Cloud so that the output from the other remaining read plugins (cpu, interface, and memory) will not be sent to Oracle Management Cloud.

Receive Metrics from a Remote Generic Metric Collector

For environments where a local Cloud agent is not installed on the host running the Generic Metric Collector, it is still possible for Oracle Management Cloud to receive the collected metrics even though it's not actually being monitored by a Cloud agent.

1. Set up integration between the collectd collector and Cloud agent to only receive metrics from the collector but not monitor it. Do NOT add the Cloud agent host as a remote host in this case.
2. Add a generic metric collector entity to the Cloud agent as shown in the following sample JSON files. Both automatic and manual metric mapping samples are shown.

Automatic collectd--Oracle Management Cloud Metric Mapping

```
{
  "entities":
  [
    {
      "name": "<Your name for the collectd collector>",
      "type": "omc_generic_metric_collector",
      "displayName": "<Your display name for the collectd collector>",
      "timezoneRegion": "<Your timezone>",
      "properties":
      {
        "host_name":
        {
          "displayName": "Host Name",
          "value": "<Your name of the host where collectd is installed>"
        },
        "omc_filter_expression":
        {
          "displayName": "Filter Expression",
          "value": "${$.[?(@.host=='<Value of the host field in the metric
payload sent by collectd>')]}"

```

```

        "displayName": "Cloud Agent Monitored",
        "value": "FALSE"
    },

    "omc_product_name":
    {
        "displayName": "Product Name",
        "value": "collectd"
    },

    "omc_product_vendor":
    {
        "displayName": "Product Vendor",
        "value": "Florian octo Forster, et al."
    },

    "omc_protocol":
    {
        "displayName": "Protocol",
        "value": "https"
    },

    "omc_payload_format":
    {
        "displayName": "Payload Format",
        "value": "json"
    },

    "omc_receiver_uri_path":
    {
        "displayName": "Receiver URI Path",
        "value": "/emd/receiver/gmc"
    }
    }
}
]
}

```

Manual collectd--Oracle Management Cloud Metric Mapping

```

{
  "entities":
  [
    {
      "name": "<Your name for the collectd collector>",
      "type": "omc_generic_metric_collector",
      "displayName": "<Your display name for the collectd
collector>",
      "timezoneRegion": "<Your timezone>",
      "properties":
      {
        "host_name":
        {
          "displayName": "Host Name",

```

```

        "value": "<Your name of the host where collectd is installed>"
    },
    "omc_filter_expression":
    {
        "displayName": "Filter Expression",
        "value": "{$.[?(@.host=='<Value of the host field in the metric
payload sent by collectd>')]}"

```

```

        "displayName": "Receiver URI Path",
        "value": "/emd/receiver/gmc"
    }
}
]
}

```

3. Configure the collectd **write_http** plugin to send metrics to the following URL:

```

https://<remote-cloud-agent-host>:<remote-cloud-agent-port>/emd/
receiver/gmc

```

There is no need to configure the *unixsock* plugin or add the Cloud agent user to the collectd socket group.

Once this configuration is complete, the Cloud agent will not try to monitor the Generic Metric Collector entity. It will instead send an *unknown* response to Oracle Management Cloud so that the availability of the Generic Metric Collector will be unknown.

Availability (Up/Down) Status for Entities Monitored by collectd

For most entities monitored by collectd, availability is turned off by default, but can be enabled by setting the appropriate `emd.properties` file property. For a specific subset of entities where the concept of “availability” does not apply, availability processing cannot be enabled.

By default, availability processing is turned on for the following collectd plug-ins:

- processes
- postgresql

Availability processing is not applicable (and cannot be enabled) for the following collectd plug-ins:

- cpu
- df
- disk
- interface
- load
- memory
- swap
- vmem

Availability processing for all other collectd plug-ins is set to “not applicable” by default, but can be enabled via the following property settings in the `emd.properties` file.

Table G-2 Availability Properties

Property	Description	Example
<code>_gmcReceiver_downResponseMultiplicationFactor</code>	Sets multiplication factor for down detection to 4 times the collection interval instead of the default value of 3.	<code>_gmcReceiver_downResponseMultiplicationFactor=4</code>
<code>_gmcReceiver_downResponseWaitTimeSeconds</code>	Sets the wait time for down response to 35 seconds while default is 30 seconds.	<code>_gmcReceiver_downResponseWaitTimeSeconds=35</code>
<code>_gmcReceiver_availabilityApplicableEntityTypes</code>	Enable sending availability (up/down) for collectd plug-ins.	The following example illustrates enabling availability processing for collectd's plugin1. <code>_gmcReceiver_availabilityApplicableEntityTypes=_gmc_collectd_plugin1</code>
<code>_gmcReceiver_responseProcessorIntervalSeconds</code>	Increases the response processor interval from the default 1 minute to 10 minutes.	The following example sets the response processor interval to 10 minutes. <code>_gmcReceiver_responseProcessorIntervalSeconds=600</code>

Troubleshooting collectd Metric Collection

If expected collectd metric data is not appearing in Infrastructure Monitoring, use the following basic debugging procedure.

1. Ensure that the generic metric collector entity was added successfully to the agent by the "omcli add_entity" command. If it is not showing up in the metric browser, run the `status_entity omcli` command.

```
$ omcli status_entity agent <entityDefinitionJsonFilePath>
```

Validation errors, if any, will be shown in the output.

2. Enable trace level logging in `emd.properties`. Set the following two properties

```
Logger._enableTrace=true
Logger.sdklog.level=DEBUG
```

and bounce the Cloud agent. Tail `gagent_sdk.trc` in agent's log directory.

3. From the log file you should see the complete payload received by agent from collectd, which metrics are in turn being sent by receiver to Oracle Management Cloud, and which metrics are unmapped. Search for "gmcReceiver received payload" in the log file to see the full payload received. If this line is not seen in the log file, the agent may not be receiving data from collectd. So check if collectd is running and that read plugins are loaded and reading metrics. To get a list of identifiers against which read plugins are collecting metrics, run the `collectdctl listval` command. To check the data source

names for each metric, run the `collectdctl getval <identifier>` command as shown below.

```
$ /opt/collectd/bin/collectdctl listval
myhost.mycompany.com/load/load
myhost.mycompany.com/processes/fork_rate
myhost.mycompany.com/processes/ps_state-blocked
myhost.mycompany.com/processes/ps_state-paging
myhost.mycompany.com/processes/ps_state-running
myhost.mycompany.com/processes/ps_state-sleeping
myhost.mycompany.com/processes/ps_state-stopped
myhost.mycompany.com/processes/ps_state-zombies

$ /opt/collectd/bin/collectdctl getval myhost.mycompany.com/load/
load
shortterm=1.100000e-01
midterm=8.000000e-02
longterm=6.000000e-02
```

Check that the `write_http` plugin has been configured correctly. Check the configured log file or `syslog` for any error message from the `write_http` plugin. Check if other software applications such as SELinux, antivirus, or a firewall may be blocking `collectd`'s ability to write metrics to the cloud agent's port.

4. Search for payload level summary lines in the log file which starts with the "**Source Metrics**" line. These lines should give a summary count of statistics such as how many metrics are being received in each payload, how many have been sent to Oracle Management Cloud, or how many are unmapped.

Payload Level Summary Logging Example - gcagent_sdk.trc

```
2017-07-04 21:45:04,613 [401336:9A108C02] DEBUG
-                               Source Metrics: 18
2017-07-04 21:45:04,613 [401336:9A108C02] DEBUG
-                               SEND_METRIC_GROUP_CALLED: 18
```

- If the summary shows **SEND_METRIC_GROUP_CALLED: <count>**, that's normal.
- If the summary shows **NO_ASSOC_GMC_ENTITY_WITH_MONITORING_CAPABILITY: <count>**, then check that `omc_filter_expression` of the generic metric collector (gmc) entity allows the payload to filter through. Ensure that the name of the host field (if any) specified in the `omc_filter_expression` property exactly matches the host field's value in the payload. Also ensure that the gmc entity has either standard or enterprise license. License can be checked from Oracle Management Cloud's Administration UI.
- If the summary shows **UNMAPPED: <count>**, then this will be accompanied by more detailed logging lines calling out the exact metrics that are unmapped. Unmapped metrics are only expected in the manual mapping case. The corresponding mapping metadata rule needs to be re-checked with the source data sent by `collectd`, particularly the filter expressions in metadata. Check if they match up with the metric data and structure.

- If the summary shows **METRIC_UPLOAD_RATE_LIMIT_EXCEEDED**: <count>, then <count> metrics in the payload were down-sampled. They were not sent to up Oracle Management Cloud. This is expected if the sending interval is anything lower than once a minute (Interval 60 in the collectd.conf file).
 - If the summary shows **WAITING_FOR_MAPPING_METADATA**: <count>, then <count> metrics in the payload are waiting for auto-map processing to complete. This is a transient state only expected in the automatic mapping case. Auto-map processing can take a few minutes to a tens of minutes to complete.
5. Further Steps - Automatic Mapping Case
 - a. If **SEND_METRIC_GROUP_CALLED**: <count> is seen, you should eventually start seeing entities on the monitoring service UI with type same as the collectd plugin name and entity name containing the collectd host's name (as obtained from the host field within the payload sent by collectd to Cloud agent). If you do not see such an entity, it's possible that the entity has been created, but has not been granted Standard or Enterprise license. This can be fixed by adding a license from the License Administration UI. From the Oracle Management Cloud console, select the *Administration > Entities Configuration > Licensing* link. From this page, look at the **Unlicensed Entities** link. If it shows the auto-created entity, assign License Edition = **Standard** or **Enterprise** and click **Save**. To ensure this happens automatically in future, set the **License Auto-Assignment** to *Standard* or *Enterprise*.
 - b. Once the auto-created entity shows up on the list of entities in the monitoring service UI, drill down into the entity to see the auto-mapped metrics. Only the availability metric will be shown by default. On the **Performance Charts** tab, Click **Options > Choose Metrics** to select the auto-created metrics for viewing their charts. Metric alert rules can also be defined on these performance metrics and are expected to work similar to alerts on metrics natively collected by Oracle Cloud agent.
 6. Further Step - Manual Mapping Case. This step is only applicable to manual mapping. If metrics are being sent from the agent to Oracle Management Cloud, check that the shape of the metrics sent (entity type, metric group and columns) are as expected and that the metrics are sent against the correct entity name. Search for the particular metric name or entity type in the log. Look for lines containing the string "Calling sendMetricGroup".

Send Metric Group Example - gcagent_sdk.trc

```
2018-03-04 11:25:04,610 [401336:9A108C02] DEBUG - Calling sendMetricGroup
on targetID=_gmc_collectd_snmp.myhost.mycompany.com;
PostMetricActionBinding [mergeKey=MergeKey
[entityName=myhost.mycompany.com, metricGroupName=if_octets_eth0,
MetricColumnWrappers=[MetricColumnWrapper [getColumnName()=rx,
isNumeric()=true], MetricColumnWrapper [getColumnName()=tx,
isNumeric()=true]], entityType=_gmc_collectd_snmp, collectionTS=Mon Mar
05 22:27:40 PST 2018], metricValues=[[6634.04566173485,
2624.41806371072]], gmcEntityName=collectd-slc11ciy-automap,
response=null, autoMapRequests=null
```

From the above log line, the target entity type is `_gmc_collectd_snmp`; entity name is `myhost.mycompany.com`; metric group name is `if_octets_eth0`; metric column names are `rx` and `tx`; metric values are `6634.04566173485` and `2624.41806371072`; and the associated gmc entity name is `collectd-slc11ciy-automap`.

7. When debugging is no longer required, turn off trace level logging and set the SDK log level to INFO. Set the following in *emd.properties*.

```
Logger._enableTrace=false  
Logger.sdklog.level=INFO
```

H

Additional Telegraf Configurations and Information

This appendix contains the following topics:

- [Metric Schema Mapping \(Telegraf\)](#)
- [Receive Metrics from a Remote Telegraf Collector](#)
- [Troubleshooting Telegraf Metric Collection](#)

Metric Schema Mapping (Telegraf)

Telegraf's metrics are auto-mapped to Oracle Management Cloud metrics.



Note:

Manual mapping of Telegraf metrics to Oracle Management Cloud metrics is not currently supported.

The following table shows how Telegraf's metric schema is automatically mapped to Oracle Management Cloud's metric schema.

Field	Value	Example
parentTargetType(64)	omc_target	omc_target
entityType(64)	'_gmc_telegraf_'<plugin>	<ol style="list-style-type: none">1. _gmc_telegraf_cpu2. _gmc_telegraf_mem
typeDisplayName(128)	<Title cased plugin name, with underscores replaced by spaces>	<ol style="list-style-type: none">1. Cpu2. Mem
entityName(256)	<host tag value> or <host tag value>'-'<unique tag value>	<ol style="list-style-type: none">1. myhost.myco.com2. myhost.myco.com-cpu0 myhost.myco.com-cpu1 myhost.myco.com-cpu2 myhost.myco.com-cpu3 myhost.myco.com-cpu-total
metricGroupName(64)	<plugin>	<ol style="list-style-type: none">1. cpu2. mem

Field	Value	Example
metricGroupDisplayName(256)	<metricGroupName in title case with underscores replaced by spaces>	<ol style="list-style-type: none"> Cpu Mem
metricColumnName(64)	<field name>	<ol style="list-style-type: none"> usage_system, usage_user used, free, total
metricColumnDisplayName(256)	<Title cased plugin name followed by a space followed by title cased field name, with underscores replaced by spaces>	<ol style="list-style-type: none"> Cpu Usage System, Cpu Usage User Mem Used, Mem Free, Mem Total
metricColumnClass	TS if the field name is "timestamp" or ends with "_ts"; STR if the field value is a string in the JSON sent by the http output plugin; NUM otherwise	NUM

You can extract the plugin, tags and field names by running the `telegraf --test` command.

Example

```
$ telegraf --test
2019/03/04 21:00:09 I! Using config file: /etc/telegraf/telegraf.conf
> cpu,collector=telegraf,cpu=cpu0,host=myhost.myco.com
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_user=0 1551762010000000000
> cpu,collector=telegraf,cpu=cpu1,host=myhost.myco.com
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_user=0 1551762010000000000
> cpu,collector=telegraf,cpu=cpu2,host=myhost.myco.com
usage_guest=0,usage_guest_nice=0,usage_idle=98.00000004470348,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_user=1.99999998952262 1551762010000000000
> cpu,collector=telegraf,cpu=cpu3,host=myhost.myco.com
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_user=0 1551762010000000000
> cpu,collector=telegraf,cpu=cpu-total,host=myhost.myco.com
usage_guest=0,usage_guest_nice=0,usage_idle=100,usage_iowait=0,usage_irq=0,usage_nice=0,usage_softirq=0,usage_steal=0,usage_system=0,usage_user=0 1551762010000000000
> mem,collector=telegraf,host=myhost.myco.com
active=6735482880i,available=11130187776i,available_percent=73.67584678266645,buffered=3569352704i,cached=7279378432i,commit_limit=22233530368i,committed_as=4000460800i,dirty=839680i,free=281456640i,high_free=0i,high_total=0i,huge_page_size=2097152i,huge_pages_free=0i,huge_pages_total=0i,inactive=5336559616i,low_free=0i,low_total=0i,mapped=1415385088i,p
```

```
age_tables=116322304i,shared=1340026880i,slab=2446262272i,swap_cached=1441792
0i,swap_free=14367285248i,swap_total=14680047616i,total=15106969600i,used=397
6781824i,used_percent=26.324153217333542,vmalloc_chunk=35184301154304i,vmallo
c_total=35184372087808i,vmalloc_used=50819072i,wired=0i,write_back=0i,write_b
ack_tmp=0i 1551762010000000000
```

The entity type is created based on the plugin, and field names in the data sent by Telegraf's HTTP output plugin. For entity naming, the value of the host tag is used primarily. In case the data contains fields that depend on additional information in the tag for uniqueness, such a unique tag value is automatically identified and appended to the entity name with a hyphen separator. You can override this behavior by adding a tag called `entity_identifier` to the input plugin. The `entity_identifier` tag specifies a custom tag that will be used to name the entity instances uniquely.

Example

```
...
# Read metrics about cpu usage
[[inputs.cpu]]
  ## Whether to report per-cpu stats or not
  percpu = true
  ## Whether to report total system cpu stats or not
  totalcpu = true
  ## If true, collect raw CPU time metrics.
  collect_cpu_time = false
  ## If true, compute and report the sum of all non-idle CPU states.
  # report_active = false
[inputs.cpu.tags]
entity_identifier = "cpu"
...
```

Limitations

Here are some of the known limitations with metric schema mapping between Telegraf and Oracle Management Cloud:

1. Field data specified in the input plugin must be uniquely identifiable using either one or two tags. That is, we require the data to be unique for each host (as resolved using the host tag's value) or made unique in combination with a single additional tag's value. For example, data from the *procstat* plugin which is configured by specifying a process name or pattern that can either match a single process or multiple processes. When it matches a single process, the resulting field data will contain a single PID record at a given timestamp and can be mapped uniquely in Oracle Management Cloud. When the match results in multiple processes, the resulting in field data with multiple PID records cannot currently be mapped to Oracle Management Cloud without data loss. For more information on this, see [Troubleshooting Telegraf Metric Collection](#).
2. The maximum number of fields per plugin that can be mapped to Oracle Management Cloud is 100. Mapping of Telegraf plugins with more than 100 fields to Oracle Management Cloud is not currently supported. For example data from Telegraf's *nstat* plugin (with over 100 fields) cannot be mapped to Oracle Management Cloud.
3. Ingestion of aggregate metrics such as *sum*, *min*, *max*, *mean*, *count*, *histograms*, etc. from Telegraf into Oracle Management Cloud is currently not supported.

4. Ingestion of metrics from plugins which emit multiple sets of fields and/or tags at a given timestamp, such as *snmp* and *procstat* is not currently supported.
5. The maximum number of metric groups per entity type (auto-created by Oracle Management Cloud when using the automatic mapping) is 50. On the Telegraf side, this translates to 50 distinct plug-ins sending data to Oracle Management Cloud.

Receive Metrics from a Remote Telegraf Collector

In environments where a local cloud agent cannot be installed on the host running the Telegraf Collector, it is still possible for Oracle Management Cloud to receive Telegraf metrics via a remote cloud agent. However, in this type of deployment, the Telegraf service itself will not be monitored by the remote cloud agent.

To configure remote monitoring:

1. Install a cloud agent on a host that is remote to the host on which the Telegraf Collector is installed
2. Configure the Telegraf Collector to send metrics to the remote cloud agent. The URL configured for the HTTP output plugin will be as follows:

```
url = "https://<remote-cloud-agent-host>:<remote-cloud-agent-
port>/emd/receiver/gmc"
```

The remaining configuration of the global collector tag and input plugins is the same as a local Telegraf configuration.

3. Add a generic metric collector entity to the cloud agent, as shown in the following example.

```
{
  "entities":
  [
    {
      "name": "<Your name for the Telegraf collector>",
      "type": "omc_generic_metric_collector",
      "displayName": "<Your display name for the Telegraf
collector>",
      "timezoneRegion": "<Your timezone>",
      "properties":
      {
        "host_name":
        {
          "displayName": "Host Name",
          "value": "<Your name of the host where Telegraf is
installed>"
        },
        "omc_filter_expression":
        {
          "displayName": "Filter Expression",
          "value": "${$.[?(@.host=='<Value of the host tag in the
metric payload sent by Telegraf>')]}"

```

```
    "omc_auto_map":
    {
      "displayName": "Automatically Map Metrics",
      "value": "TRUE"
    },
    "capability":
    {
      "displayName": "capability",
      "value": "monitoring"
    },
    "omc_monitored":
    {
      "displayName": "Cloud Agent Monitored",
      "value": "FALSE"
    },
    "omc_product_name":
    {
      "displayName": "Product Name",
      "value": "telegraf"
    },
    "omc_product_vendor":
    {
      "displayName": "Product Vendor",
      "value": "InfluxData Inc."
    },
    "omc_protocol":
    {
      "displayName": "Protocol",
      "value": "https"
    },
    "omc_payload_format":
    {
      "displayName": "Payload Format",
      "value": "json"
    },
    "omc_receiver_uri_path":
    {
      "displayName": "Receiver URI Path",
      "value": "/emd/receiver/gmc"
    }
  }
}
]
```

In this case, the host where Telegraf is installed as specified in the `host_name` property will not be a managed target in Oracle Management Cloud.

With the above this configuration, when Telegraf is started up, the cloud agent will be able to receive metrics from the remote Telegraf Collector without monitoring the Telegraf service. The availability of the Generic Metric Collector entity representing the Telegraf Collector in this case will be reported as *unknown*.

Availability (Up/Down) Status for Entities Monitored by Telegraf

For most entities monitored by Telegraf, availability is turned off by default, but can be enabled by setting the appropriate `emd.properties` file property. For a specific subset of entities where the concept of “availability” does not apply, availability processing cannot be enabled.

Availability processing is not applicable (and cannot be enabled) for the following Telegraf plug-ins:

- `cpu`
- `disk`
- `diskio`
- `interface`
- `kernel`
- `mem`
- `swap`
- `system`

Availability processing for all other Telegraf plug-ins is set to “not applicable” by default, but can be enabled via the following property settings in the `emd.properties` file.

Table H-1 Availability Properties

Property	Description	Example
<code>_gmcReceiver_downResponseMultiplicationFactor</code>	Sets multiplication factor for down detection to 4 times the collection interval instead of the default value of 3.	<code>_gmcReceiver_downResponseMultiplicationFactor=4</code>
<code>_gmcReceiver_downResponseWaitTimeSeconds</code>	Sets the wait time for down response to 35 seconds while default is 30 seconds.	<code>_gmcReceiver_downResponseWaitTimeSeconds=35</code>
<code>_gmcReceiver_availabilityApplicableEntityTypes</code>	Enable sending availability (up/down) for Telegraf plug-ins.	The following example illustrates enabling availability processing for Telegraf’s <code>plugin2</code> . <code>_gmcReceiver_availabilityApplicableEntityTypes=_gmc_telegraf_plugin2</code>

Table H-1 (Cont.) Availability Properties

Property	Description	Example
_gmcReceiver_responseProcessorIntervalSeconds	Increases the response processor interval from the default 1 minute to 10 minutes.	The following example sets the response processor interval to 10 minutes. <pre>_gmcReceiver_responseProcessorIntervalSeconds=600</pre>

Custom Metric Collection Methods and Metric Columns

Collection methods provide comprehensive, easy-to-use monitoring connectivity with a variety of target types. Metric columns define the data returned by the collection method.

Collection Method

A collection method enables communication with a specific entity type and translates the entity data to standards-compliant XML and back. The custom metric entity type determines which collection methods are made available from the Oracle Management Cloud console. For example, when creating a custom metric for an Automatic Storage Management entity type, only two collection methods (OS Command and SQL) are available from the UI.

Oracle Management Cloud provides the following collection methods:

- [OS Command](#)
- [SQL Query](#)
- [Java Management Extensions \(JMX\)](#)
- [REST](#)

Metric Columns (Advanced Options)

Metric columns define the data that is returned by the collection method. In certain situations, you may want the metric column value to be based on calculations performed using other metric columns, or perhaps differences and/or rates of change between metric columns. Oracle Infrastructure Monitoring allows you to perform advanced operations on metric columns to increase metric utility and flexibility:

- [Compute Expressions](#)
- [Rate and Delta Metric Columns](#)

OS Command

This collection method executes the specified command and returns the command output, delimited by a specified string, as multiple columns.

For example, if the command output is: `em_result=1|2|3` and the Delimiter is set as `|`, then three columns are populated with values 1,2,3 respectively.

Properties

- **Command** - The command to execute. For example, `%perlBin%/perl`. The complete command line will be constructed as: *Command + Script + Arguments*.
- **Script** - A script to pass to the command. For example, `%scriptsDir%/myscript.pl`. You can upload custom files to the agent, which will be accessible under the `%scriptsDir%` directory.

- Script Location - The absolute path to the script. This path and script location must be accessible by the cloud agent user.
- Arguments - Additional arguments required by the script.
- Prefix for Output - The starting string of metric result lines. For example, if the command output is: `em_result=4354 temp res` you can set "Starts With" = `em_result`, so that only lines starting with `em_result` will be parsed.
- Delimiter for Output - The string used to delimit the command output.

Available Variables

Variables can be used in collection method properties. Variable names are case-sensitive. To escape '%', use '%%'.

Name	Description
%perlBin%	Location of perl binary.
%scriptsDir%	Directory where scripts are stored.
%NAME%	Name of the entity.
%TYPE%	Entity type.
%DISPLAY_NAME%	Display name of target instance.
%TYPE_DISPLAY_NAME%	Display name of target type.

SQL Query

The SQL Query collection method allows you to execute a normal SQL query or PL/SQL statement against the database to retrieve data.

Properties

- SQL Query - The SQL query to execute. Normal SQL statements should not be semi-colon terminated. For example, `SQL Query = "select a.ename, (select count(*) from emp p where p.mgr=a.empno) directs from emp a"`. PL/SQL statements are also supported, and if used, the "Out Parameter Position" and "Out Parameter Type" properties should be populated.
- SQL Query File - A SQL query file. Note that only one of "SQL Query" or "SQL Query File" should be used. For example, `%scriptsDir%/myquery.sql`. You can upload custom files to the agent, which will be accessible under the `%scriptsDir%` directory.
- Out Parameter Position - The bind variable used for PL/SQL output. Only a number should be specified. For example, if the SQL Query is:

```

DECLARE
    l_output1 NUMBER;
    l_output2 NUMBER;
BEGIN
    .....
    OPEN :1 FOR
        SELECT l_output1,l_output2 FROM dual;
END;
```

then you can set Out Parameter Position = 1, and Out Parameter Type = SQL_CURSOR

- Out Parameter Type - The SQL type of the PL/SQL output parameter.

Available Variables

Variables can be used in collection method properties. Variable names are case-sensitive. To escape '%', use '%%'.

Name	Description
%perlBin%	location of perl binary
%scriptsDir%	directory where scripts are stored
%NAME%	name of target instance
%TYPE%	target type
%DISPLAY_NAME%	display name of target instance
%TYPE_DISPLAY_NAME%	display name of target type
%OracleHome%	Oracle Home Path
%MachineName%	Listener Machine Name
%Port%	Port
%SID%	Database SID

Java Management Extensions (JMX)

This collection method can be used to retrieve JMX attributes from JMX enabled servers and returns these attributes as a metric table.

Properties

- MBean Name - This is the MBean ObjectName or ObjectName pattern whose attributes are to be queried. Since this is specified as metric metadata, it needs to be instance agnostic, so instance specific key-properties if any (like servername), on the MBean ObjectName may need to be replaced with wild-cards.
- JMX Attributes - This is a semicolon separated list of JMX attributes in the order they need to be presented in the metric.
- Identity Column - This is an MBean key property that needs to be surfaced as a column when it is not available as a JMX attribute.

Example: `com.myCompany:Name=myName,Dept=deptName, prop1=prop1Val, prop2=prop2Val`

In this example, setting `identityCol` as `Name;Dept` will result in two additional key columns representing `Name` and `Dept` besides the columns representing the JMX attributes specified in the `Column Order` property above.

- Auto Row Prefix - This is prefix used for an automatically generated row, in case the MBean ObjectName pattern specified in metric property matches multiple MBeans and none of the JMX attributes specified in the `Column Order` are unique for each. The Auto Row Prefix value specified here will be used as a prefix for the additional key column created.

Example: If the Metric is defined as `com.myCompany:Type=CustomerOrder,*`

`Column Order` is `CustomerName;OrderNumber;DateShipped`

and assuming `CustomerName;OrderNumber;Amount` may not be unique

if an order is shipped in two parts, setting `Auto Row Prefix` as

"ShipItem-" will populate an additional key column for the metric

for each row with

ShipItem-0,

ShipItem-1

- **Metric Service - True/False.** This indicates whether Metric Service is enabled on the target WebLogic Server domain. If set to true, then the basic property MBean Name above should be the Metric Service table name and the basic property JMX Attributes should be a semicolon-separated list of column names for above Metric Service table.

Available Variables

Variables can be used in collection method properties. Variable names are case-sensitive. To escape '%', use '%%'.

Name	Description
%perlBin%	Location of perl binary
%scriptsDir%	Directory where scripts are stored
%NAME%	Name of target instance
%TYPE%	Target type
%DISPLAY_NAME%	Display name of target instance
%TYPE_DISPLAY_NAME%	Display name of target type
%ServerNames%	Server Names
%version%	Version
%ObjectName%	Object Name
%OracleHome%	Oracle Home
%ProxyMBeanObjectName%	Proxy MBean ObjectName
%OracleInstance%	Oracle Instance
%CanonicalPath%	Canonical Path
%compVersion%	Component Version
%OPMNMBeanName%	OPMN MBean Name
%VersionCategory%	Version Category

REST

The REST (Representational State Transfer) collection method retrieves attributes from enabled servers and returns the results as a table.

Properties

- **Namespace - Set of all namespaces referenced. Specify using notation:** [ns0="uri0"][ns1="uri1"].. Example: [ns0="http://type.abc.com"][ns1="http://app.abc.com"]
- **Column Type - List all the metric column names and their types. Supported types are STRING and NUMBER. The order and data type of the columns should match the order and data type of the columns listed in the extension.**

Example

If the response payload (XML) is:

```
<ns0:getEmployeeDataResponse xmlns:ns0="http://sample.demo.com">
  <employee>1234</employee>
  <id>1234</id>
  <title>4</title>
</ns0:getEmployeeDataResponse>
```

The Column Type will be:

```
Employee:STRING,Title:STRING, ID:NUMBER
```

 **Note:**

The order of columns in Column Type should match the order of columns in the Row Type property.

- Row Type - Provide a XPath/JSON-Path expression corresponding to each metric column defined above. For multiple metric columns, they should be separated by commas.

Example

If the response payload (XML) is:

```
<ns0:getEmployeeDataResponse xmlns:ns0="http://sample.demo.com">
  <employee>1234</employee>
  <id>1234</id>
  <title>4</title>
</ns0:getEmployeeDataResponse>
```

The Row Type will be (in XPath):

```
/ns0:getEmployeeDataResponse/employee,/ns0:getDataResponse/title,/
ns0:getDataResponse/id
```

- Request element payload - Payload Element in XML/JSON format. Must be specified using the CDATA section if it is XML. (optional)
- Request Metadata - A serialized string of the object *oracle.sysman.emInternalSDK.webservices.rs.api.request.Resource*

Available Variables

Variables can be used in collection method properties. Variable names are case-sensitive. To escape '%', use '%%'.

Name	Description
%scriptsDir%	Directory where scripts are stored.
%NAME%	Name of the entity.
%TYPE%	Entity type.
%DISPLAY_NAME%	Display name of target instance.

Name	Description
%TYPE_DISPLAY_NAME%	Display name of target type.

Compute Expressions

You use compute expressions to calculate the value of a metric column based on mathematical or logical operations performed on other metric columns.

Compute expressions require at least one other metric column to be defined first, and can only include those metric columns that are listed before this metric column in order. You can use the up and down arrows to re-order metric columns. The value of the column is calculated using the given compute expression.

The following table shows operators which can be used while defining compute expression.

Operator	Example	Explanation
+	Column1 + Column2	Returns the sum of the values of Column1 and Column2.
-	(Column1 + Column2) - Column3	First add Column1 and Column2 values, then subtract Column3 value and return the result.
*	(Column1*Column2) + Column3	First multiply Column1 and Column2 values, then add Column3 value and return the result.
/	(Column1 + Column2) /2	Returns the average of Column1 and Column2 values.
__ceil	__ceil Column1	Returns the value of Column1 rounded off to the largest integer.
__floor	__floor Column1	Returns the value of Column1 rounded off to the lowest integer.
__round	__round Column1	This expression will round the value of Column1 to the nearest integer, away from zero.
==	Column1 == 1	Returns true if the value of Column1 is 1, else returns false.
!=	Column1 != 1	Returns false if the value of Column1 is 1, else returns true.
() ? : ;	(Status == 1) ? "UP" : "DOWN"	This operator is equivalent to if then else statement. This expression will return "UP" if Status value is 1 otherwise it will return "DOWN"

Operator	Example	Explanation
<code>__is_null</code>	<code>__is_null Column1</code>	Returns true if the value of Column1 is NULL, else returns false.
<code>__delta</code>	<code>__delta Column1</code>	Returns the difference between the current value and the previous value of Column1.
<code>__contains</code>	<code>Column1 __contains "ORA-"</code>	Returns true if the value of Column1 contains the string "ORA-", else returns false.
<code>__beginswith</code>	<code>Column1 __beginswith "ORA-"</code>	Returns true if the value of Column1 starts with the string "ORA-", else returns false.
<code>__matches</code>	<code>Column1 __matches "UP"</code>	Returns true if the value of Column1 is equal to "UP", else returns false.
<code>__length</code>	<code>__length Column1</code>	Returns the length of string value of Column1.
<code>__to_upper</code>	<code>__to_upper Column1</code>	Returns the upper case of string value of Column1
<code>__to_lower</code>	<code>__to_lower Column1</code>	Returns the lower case of string value of Column1.
<code>__interval</code>	<code>Column1 / __interval</code>	Returns the Column1 value divided by the collection interval.

Refer to the examples for details about the expression grammar and usage.

Value	Definition
<code>__interval</code>	Collect interval.
<code>__sysdate</code>	Current system time.
<code>__GMTdate</code>	Current GMT time.
<code>__contains</code>	Tests a given string expression for presence of a string expression.
<code>__beginswith</code>	Tests whether a given string expression begins with a specified string expression.
<code>__endswith</code>	Tests whether a given string expression ends with the specified string expression.
<code>__matches</code>	Tests whether a given string expression matches a specified string expression.
<code>__delta</code>	Computes the difference between the current value and the previous value.
<code>__leadingchars</code>	Returns the leading characters in the specified string.
<code>__trailingchars</code>	Returns the trailing characters in the specified string.
<code>__substringpos</code>	Returns the position of the occurrence of the pattern within a specified string.
<code>__is_null</code>	Tests whether the expression is NULL
<code>__length</code>	Returns the length of the string expression.

Value	Definition
<code>__to_upper</code>	Converts the string to upper case.
<code>__to_lower</code>	Converts the string to lowercase.
<code>__ceil</code>	Returns the smallest integral value not less than identifier.
<code>__floor</code>	Returns the largest integral value not greater than the identifier.
<code>__round</code>	Rounds to nearest integer, away from zero.

Examples:

- The value of the column is the average of the columns **Column1** and **Column2**.

```
NAME="Average" COMPUTE_EXPR="(Column1 + Column2 )/ 2"
```

- The value of the column **Version** is computed as 7.X if column **Column1** contains the String NetApp Release 7..

```
NAME="Version" COMPUTE_EXPR="(Column1 __contains 'NetApp Release 7.') ? '7.X':'6.X'"
```

- The value of the column **Column1** is the difference of the columns **Column2** and **Column3**.

```
NAME="Column1" COMPUTE_EXPR="(Column2 - Column3)"
```

- The value of the column Status is 1 if the value of column State matches the String STARTED and 0 otherwise.

```
NAME="Status" COMPUTE_EXPR="State __matches 'STARTED'"
```

- The value of the column Column1 is yes if the value of column Column2 is null and no otherwise.

```
NAME="Column1" COMPUTE_EXPR="(__is_null Column2)?'yes':'no'"
```

- The value of the column Source is lanplus if the length of string value of column result is 0; else it is the value of the column result.

```
NAME="Source" COMPUTE_EXPR="((__length result) == 0) ? 'lanplus' : result"
```

- The value of the column Rate is the value of column Column1 divided by the collection interval, rounded up to the largest integer.

```
NAME="Rate" COMPUTE_EXPR="(__ceil (Column1/__interval))"
```

- The value of the column is the Column1 when Column2 and Column3 are existing metric columns.

```
NAME="Column1" COMPUTE_EXPR="((Column2 == 0) ? 0 : ((Column3 / (Column2 / 8)) * 100.0))"
```

- The value of the column is the total percentage of disk available where Column1 and Column2 are existing metric columns

```
NAME="PERCENTAGE_VALUE" COMPUTE_EXPR="(Column1 != 0) ? 100.0*(Column2/Column1) : 0"
```

Rate and Delta Metric Columns

You can create additional metric columns based on an existing data column that measure the rate at which data changes or the difference in value (delta) since the last metric collection.

After at least one metric column has been created and the metric column row is selected in the table, two additional options appear in the **Add** menu:

- *Delta metric column on <selected metric column>*
- *Rate (per min) metric column on <selected metric column>*

To create a rate/delta metric column, click on an existing data column in the metric columns table and then select one of the rate/delta column menu options from the **Add** menu.

Usage Examples

- Add Delta metric columns based on another metric column

Example: You want to know the difference in the table space used since the last collection.

Delta Calculation:

```
current metric value - previous metric value
```

- Add Rate Per Minute metric column based on another metric column

Example: You want to know the average table space usage per minute based on the table space column metric which is collected every 1 hr.

Rate Per Minute Calculation:

```
(current metric value - previous metric value) / collection schedule
```