CHAPTER 1

Overview of Cloud Control Architecture
Oracle Enterprise Manager Cloud Control 12c (hereafter referred to as Cloud Control, CC, or CC 12c) is Oracle’s solution for managing your complete IT environment—Oracle and non-Oracle products alike. Cloud Control gathers information about your enterprise computing systems, consolidating their management in a central repository. Cloud Control displays this information to administrators from its Web console and sends them alerts on threshold conditions of interest. Administrators can then use this information to have Cloud Control perform tasks for the computing systems it monitors.

Cloud Control is built upon the Oracle technology stack, an Oracle database back end, and a WebLogic Server middle tier that hosts the Oracle Management Service (OMS) application. In a nutshell, an Oracle database hosts the Oracle Management Repository (OMR), which stores alert and metric data for managed targets that is collected and uploaded by local Oracle Management Agents (Agents or OMA). These Agents upload the data to the OMS, which in turn uploads to the Repository. The OMS also renders a Web-based Console for CC administrators and brokers their communications with the Repository. To help you better understand the architecture of Grid Control (GC) beyond this high-level description, so as to architect a Cloud Control infrastructure to meet your needs, this chapter aims to answer the following questions:

- What are the principal architectural developments from Oracle Enterprise Manager (OEM or EM) Grid Control 10g and 11g to OEM Cloud Control 12c? This will help orient those familiar with earlier EM releases to the new CC design.
- What are the main Cloud Control 12c components and their function?
- What is the interaction between CC components to allow them to work together to render the Console and process alerts and metric data from monitored targets?

Let’s begin with key architectural developments in OEM from the time of GC 10g, through GC 11g and Cloud Control 12c. While this discussion is primarily for the benefit of readers familiar with GC 10g and 11g, it will also give new Cloud Control users some background on the OEM product, and on how the new 12c release is distinguished from earlier releases.

**Architectural Developments from GC 10g to CC 12c**

The architecture and installation of Oracle Enterprise Manager GC 10g, GC 11g, and CC 12c differ for all components, and particularly for the OMS and Management Repository:
Chapter 1: Overview of Cloud Control Architecture

- **OMS Architecture and Installation**  Both the architecture and installation of the OMS diverge between GC 10g, GC 11g, and CC 12c.

  - The 11g and 12c Oracle Management Service (OMS) is built on Oracle WebLogic Server (WLS) 11g Release 1 (version 10.3.5), whereas the 10g OMS runs on Oracle Application Server 10g (10.1.2.3). This architectural distinction results in an entirely new and vastly improved Console look-and-feel in CC 12c than in GC 11g/10g.

  - The GC 11g installation on all platforms except Linux 32-bit requires that you first install Java Development Kit (JDK) 1.6, then install WLS 10.3.2 and apply patch ID WDJ7, whereas on all CC 12c platforms, the Installer takes care of WLS 10.3.5 installation and patching for you. CC 12c installs WLS, just as GC 10g lays down Oracle Application Server 10g.

- **Management Repository Installation**  As of GC 11g, the installation of the database for the Repository was decoupled from the OEM installation itself. In GC 10g, one of the Oracle Universal Installer (OUI) options was, in addition to installing the OMS, to create a new 10g Database for the Repository. By contrast, the GC 11g and 12c software distributions do not include a built-in Oracle database for housing the Management Repository; you must preinstall a database in all cases. However, CC 12.1.0.2 offers a template in Database Configuration Assistant (DBCA) that creates a preconfigured Repository database.

The following section provides more details on these key architectural and installation differences between the OEM 10g, 11g, and 12c releases.

**OEM 11g and 12c Use WLS  Whereas OEM 10g Uses AS 10g**

Moving from Oracle Application Server 10g in GC 10g to an Oracle Fusion middleware WLS platform in OEM 11g and 12c was a radical departure for Oracle, but in line with what is now the foundation of their application infrastructure. Oracle acquired BEA Systems, Inc. for 8.5 billion dollars in April 2008 as a strategic move to overtake IBM's WebSphere platform. Oracle was looking to become the service-oriented architecture (SOA) market leader, by providing complementary middleware products, Oracle OC4J Application Server and BEA WebLogic. Oracle positions WLS 11g as the industry’s most comprehensive platform for developing, deploying, and integrating enterprise applications.\(^1\) WLS is indeed the top-rated application server in the industry.\(^2\)

---


\(^2\) A recent Gartner worldwide application server market share report shows Oracle at 44 percent, higher than its four closest competitors combined.
In GC 10g, the OMS is a Java 2 Platform Enterprise Edition (J2EE) middle-tier application that renders the user interface for the GC Console. (J2EE is an environment for developing and deploying enterprise applications.) The GC 10g middle tier, which runs on Oracle Application Server (OAS or Oracle AS) 10g, contains three elements: Oracle Application Server Containers for J2EE (OC4J), Oracle HTTP Server (OHS), and OracleAS Web Cache. The OHS deploys the 10g Management Service J2EE Web application. The OMS 10g is technically part of the OC4J, but the entire GC 10g middle tier is usually referred to as the OMS. OracleAS Web Cache provides an additional way to log in to the GC Console.

In contrast to GC 10g, the GC 11g and CC 12c middleware platforms consist of a WLS instance in which an OMS application domain is created (called the GCDomain), rather than the OMS being deployed in its own OC4J container. In addition, OracleAS Web Cache is not used in GC 11g and CC 12c, as it is in GC 10g. This is a boon for GC 11g, in our opinion, for at least three reasons. The first is that Web Cache is not terribly useful in GC 10g, and provides very little performance advantage over logging in to the Console directly to OHS. Most Console requests, being ad hoc, are for dynamic data with very little cached content, generally limited to icons, menu items, headers, and footers. Secondly, Web Cache complicates the diagnosis of GC 10g problems, as many Enterprise Manager analysts at Oracle Support can surely attest. Lastly, Console access via Web Cache is unsecure (on HTTP port 7777) out of the box, and the configuration process to secure such access is nontrivial and undocumented. This is a gaping security hole for sites that want to use Web Cache, but need to enforce secure communications between all GC components.

**NOTE**

Web Cache in GC 10g acts like a virtual server on behalf of OHS for the Management Service. If any part of the content is in its cache, such as from navigating to a previous page, Web Cache sends that part of the content directly to the Console browser and stores a copy of the page in cache. This is known as a cache hit. If Web Cache does not have the requested content, or if the content is stale or invalid, it hands off the request to the OHS, which requests the data from the Management Service. This is known as a cache miss.

Oracle’s direction to use WLS rather than OAS as the middleware platform for the OMS cannot be disputed from the standpoint of technology choice. By all accounts, WLS is a superior product to OAS 10g in many respects. Oracle could have taken the direction of using Oracle Application Server (AS) 11g for the OMS application. However, AS 11g itself uses WLS as its J2EE component. The technological direction
came down to whether to build the OMS application in an OC4J container (used in GC 10g) or in a WLS instance. While Oracle provides customers with the choice to build their own applications on either OC4J or WLS, it is clear which choice Oracle itself preferred for its own applications, GC 11g and CC 12c.

**Must Preinstall a Database for the Repository in OEM 11g and 12c**

Before installing GC 11g or CC 12c, you must first install a certified database to house the Management Repository. By contrast, in GC 10g, while you had the option to first manually install an Oracle 10g or 11g Database, you could alternatively choose to have the OUI install a single-instance Oracle 10g Database containing the Management Repository. This distinction is more than just a superficial difference in installation choices; it is an architectural diversion between 10g and the later 11g and 12c releases. One can safely assume that the Oracle EM development team recognized that there are too many supported releases (including patch sets) and types of Oracle database installations available in OEM 11g and 12c to be able to choose just one type or offer multiple installation types through the OEM Installers. Database releases and patch sets include 11.2.0.1+, 11.1.0.7, and 10.2.0.5+, and installation types include Oracle RAC, single-instance, Oracle RAC One-Node (ORON), and Cold Failover Cluster (CFC). The EM development team understandably wanted to get out of the business of having to maintain links in the OEM Installer to one or more Oracle database releases and patch sets certified for the Management Repository.

Even if the EM group chose for the 11g and 12c Installers only to offer the latest certified database release and patch set for the Repository, this would be a moving target. That is, new database patch sets released and certified with OEM would supersede any earlier database patch set that might be bundled with the OEM software. By not offering to lay down an Oracle database as an option in the OEM 11g or 12c Installers, the responsibility for selection of a particular Oracle database release, and the database architecture, is left with the DBA performing the EM installation. In our view, this is a good direction for Oracle to have taken, particularly now that CC 12.1.0.2 offers a template for DBCA that creates and configures an 11.2.0.3 Database and preinstalls the Repository in it, before the Cloud Control Installer is run to install the OMS. It was certainly convenient for customers to let the GC 10g Installer lay down all GC components, including a database for its Repository. However, we observed that many customers who installed GC 10g with the New Database Installation Type never patched, reconfigured, or tuned the seed 10g Database that the GC Installer dropped down. These customers made an arguably justifiable assumption that the 10g Database installed by GC 10g would be patched, configured, and tuned out of the box to support the accompanying Management Repository. Sadly, however, this was not the case. In fact, if you selected the GC New
Database Option in any release or version of GC 10g (10.1 or 10.2), the Installer laid down Database 10.1.0.4 for the Repository. This was the case long after 10.2 Database was certified with GC 10g—and even after 11.1 Database was certified. Thus, to take advantage of bug fixes and new features afforded by Database Release 10.2 over Release 10.1, customers who chose to let the GC 10g OUI install a new database had to subsequently upgrade the Repository database software and database itself from 10.1.0.4 to 10.2.0.1 (including performing post-installation database steps), and then apply the latest Database 10.2.0.x Patch Set.

Cloud Control 12c Components

Now that you have an idea of how Oracle Enterprise Manager 10g, 11g, and 12c differ architecturally, let’s examine the main components, which are common to all releases. The basic Cloud Control topology consists of four core components, as depicted in Figure 1-1: the Cloud Control Console, the Oracle Management Agent, the Oracle Management Service, and the Oracle Management Repository. Each component can be separated by a firewall.

![Cloud Control Core Components Diagram](related_diagram_url)
Chapter 1: Overview of Cloud Control Architecture

Following is a brief description of each core component.

■ **Cloud Control Console** The Cloud Control Console is a browser-based application through which administrators can centrally manage their entire computing environment. If you had to choose one image to define the big picture for Cloud Control, it would have to be the Console home page:

The home page displays an overall view of your managed IT infrastructure from which you can drill down to a specific managed target that Cloud Control administers. The CC Console is certified to run on all popular browsers, including Internet Explorer, Firefox, Safari, and Google Chrome, and to use the Adobe Flash Player plug-in for certain Console functionality. You don’t install the Console; the Management Service renders it. You just open a browser and connect to the Console via the Enterprise Manager login URL.
Oracle Management Agent  The Management Agent, installed on each managed host, monitors the host and all targets on that host, and communicates information about these targets to the OMS. Targets can be Oracle and non-Oracle components installed on the host. Cloud Control monitors over 200 different target types; each instance of a particular target type counts as a monitored target. Examples of commonly used target types are Database Instance, Listener, Oracle Application Server, and Host.

Management plug-ins permit monitoring of specific Oracle and non-Oracle target types. In Grid Control 10g and 11g, certain target types bundled with the product are now packaged as plug-ins in CC 12c. This allows the Oracle EM development team to update plug-in software for target types independently of the CC release. Plug-ins are both Oracle-built and partner-built. Oracle-built plug-ins exist for Oracle products, of course, but also for many non-Oracle products, including Microsoft SQL Server, Microsoft Active Directory, NetApp Filer, and IBM WebSphere Application Server. Partner-built plug-ins are available for other non-Oracle components typically found in today’s data centers, including F5 BIG-IP LTM, Citrix Presentation Server, and Brocade ServerIron System.

Cloud Control monitors itself, so an Agent also runs on all nodes hosting the Oracle Management Service and Management Repository. Each managed host (or virtual host) runs one and only one Agent. (Agents in Cold Failover Cluster [CFC] environments run multiple Agents, one for each cluster node, and one for the virtual host itself.) You can have as many Agents as you can scale the Cloud Control infrastructure to support, which is practically capped only by network speeds as limited by geography. Oracle certifies the Agent for the latest current 12c release (12.1.0.2) on the most common 32-bit and 64-bit host platforms, including Linux x86-64 and x86 (RHEL, OEL, SLES, and Asianux), Oracle Solaris on SPARC (64-bit), Oracle Solaris on x86-64 (64-bit), IBM AIX on POWER Systems (64-bit), IBM Linux on System z Windows (x64 and x86), and HP-UX (Itanium, PA-RISC).

Oracle Management Service  The OMS is a Java 2 Platform Enterprise Edition (J2EE) middle-tier application that renders the user interface for the Console. Agents upload target-related data to the OMS, which then processes this data before uploading it to a data store, the Oracle Management Repository. The Cloud Control middle tier consists of an Oracle WebLogic Server 10.3.5 instance, which deploys the Management Service J2EE Web application.

You must install the OMS on at least one or more hosts as needed to support your environment for scalability or high availability. Each Management Service must reside on its own host. The OMS and Management Repository
can reside on the same host, but for performance reasons, Oracle does not recommend this configuration for a production Cloud Control environment, unless it is small (fewer than 1,000 targets). All physical OMS hosts independently provide the generic Oracle Management Service, though multiple OMS hosts with a Shared Directory can coordinate to process Agent upload files to this directory.

At the time of this writing, the OMS was certified to run on Linux x86-64 and x86, Oracle Solaris on SPARC (x64), Oracle Solaris on x86-64 (64-bit), IBM AIX on POWER Systems, and Windows x86-64. Plans are to certify the OMS on Windows x86 and HP-UX (Itanium, PA-RISC), which are already certified for the Agent.

**Oracle Management Repository** The Repository (OMR) is the data store for Cloud Control, created either during the CC installation in a preinstalled Oracle database or by the EM template in DBCA when creating the database. The Repository is located in the SYSMAN schema, which contains information on all Cloud Control targets and administrators. The Repository organizes this data so that the OMS can retrieve and display it in the Console for any administrator with privileges to view it. A Cloud Control infrastructure uses just one central Repository database. It can be a single-instance or RAC Oracle database, releases 10gR2 (10.2.0.5), 11gR1 (11.1.0.7.0), or 11gR2 (11.2.0.1+), although Oracle Database 11.2 is recommended.

Console administrators and Agents communicate with the OMS, via ad hoc and batch connections (except alerts), using the following network protocols, respectively:

- An administrator requests content in the Console over HTTP(S) in a browser session, which the OMS renders. The OMS then retrieves the data for the request from the Management Repository and displays it in the Console.

- Agents upload information to the OMS over HTTP(S), and the OMS uploads this data via thin JDBC to the OMR. The OMR sends data back to the OMS over thin JDBC, which is relayed to the Agent via a built-in HTTP listener.

---

3 In earlier versions of GC 10g, it was not supported to install the OMS on a host running Oracle RAC or third-party clusterware. However, in OEM 11g and 12c, you can install the OMS on one or more RAC nodes hosting the OMR database.
Now that you are familiar with the main components of the Cloud Control architecture, let’s examine these components more closely, including their directory structure. We will end by tracing the data flow more closely between components for Console and Agent communications.

**Cloud Control Console**

The Console provides the user interface to the Cloud Control product. From any location with Web browser access, you can log in to the Console and centrally manage your entire enterprise grid environment. Because the Console interface is rendered in HTML, it uses HTTPS by default (or HTTP if unsecured), making it lightweight, easy to access, and firewall friendly.4

The usual way to log in to the Console is at http[s]:<OMShost>:<port>/em. The CC installation limits Console browser access to secure communications (SSL), but, after installation, also allows you to open up unsecure (HTTP) access with minimal configuration. (In GC 10g, securing Console communications after installation required the reverse manual configuration.)

In GC 10g, you could log in via OracleAS Web Cache, which handed you off to Oracle HTTP Server (OHS), or more directly via OHS. In OEM 11g and 12c, however, you can only log in to the Console through the WLS Apache Web server on default HTTPS port 7799. There is no Web Cache access to the 11g or 12c Console.

In GC 10g, almost all of the capability of Oracle Enterprise Manager 9i Java Console was converted to HTML within the GC 10g Console. To execute the few leftover 9i fat client features not yet available in the GC 10g Console, such as management of Oracle Advanced Replication, you had to download the Oracle10g Client software and install the Enterprise Manager 10g Java Console nondefault component. The Enterprise Manager Java Console only runs in standalone mode; it does not connect to the Management Service. As of GC 11g, there is no longer any need for a fat client Java Console. There are no OEM 11g or 12c features outside of those accessible through the browser Console.

**Oracle Management Agent**

The Management Agent monitors all Cloud Control targets on a host, including the host itself. It is installed either as a standalone Agent using one of six methods (see Chapter 5) or as part of the first OMS installation, which is known as a “chained” installation. The directory structure of the resulting Agent installation is the same, regardless of the installation method chosen or used (that is, whether chain installed or not), except for a Shared Agent (or NFS Agent), where the Agent binaries are shared

---

4 EM 10.2 offered wireless device access to Enterprise Manager functionality through its wireless component, EM2Go. However, EM2Go support ended after GC 10.2.0.2. In CC 12c, however, you can now log in to the EM full-featured desktop version directly from the Safari browser on any iDevice (iPhone, iPad, or iPod touch). The Cloud Control Mobile app is also available for iDevices as a free app from the iTunes App Store. However, CC Mobile functionality currently provides access only to Incident Manager.
on an NFS file system. The Agent is the distributed portion of the Enterprise Manager framework, and is implemented in the C programming language for performance and resource reasons. It is a multithreaded process that uses Oracle core libraries, the Oracle Call Interface (OCI), and Oracle Secure Sockets Layer (SSL) to secure it by default. A reasonable approach to an overview of the Management Agent architecture is to grasp the concept of Agent target discovery and to look at the Agent file and directory structure.

**Target Discovery and Management**

You run one and only one Management Agent on each host that has targets to be managed. A properly installed Agent automatically starts monitoring itself, its host, the OMS, if installed along with the Agent on the host, and certain Oracle products (for example, databases, listeners, and Oracle Application Servers) installed on the host. (The Agent and host are targets in their own right; Cloud Control treats them like any other target.) This automatic target discovery begins as soon as the Agent is installed and starts up. Those targets not automatically discovered can be manually discovered in the Console, and new targets can be automatically or manually discovered as well.

Targets are categorized by target type. Cloud Control manages approximately 200 out-of-box target types, and many more target types provided by plug-ins not configured with the OMS. A fair number of plug-ins for GC 11g have not yet been updated for CC 12c, but it is likely that these plug-ins will be updated in the near term.

The Management Agent uses default monitoring and data collection levels to provide monitoring and management information on discovered targets. The Agent immediately uploads metric alerts and periodically uploads management information to the OMS. The Agent also performs tasks on behalf of the OMS, from running jobs (units of work defined to automate administrative tasks such as backups or patching) to setting blackouts (suspending data collection on targets to perform scheduled maintenance). Blackouts allow for a more accurate picture of a target’s performance because Agent downtime can be characterized as planned, thereby not adversely affecting that target’s Service Level Agreement (SLA).

**Management Agent Files and Directories**

The Agent stores its configuration and management information about targets in flat files under the Agent installation base directory (AGENT_BASE), such as /u01/app/oracle/agent/. This directory holds distinctly different information in multiple Homes, for the Agent software, Agent State Home, plug-ins, and setuid binaries. Following is a description of each of these Homes, their default locations, and notable files within them:

---

5 Much of this information on Agent directory structure is taken from MOS Bulletin 1386113.1.
■ **AGENT_HOME**  $AGENT_BASE/core/12.1.0.x.0/. Contains all binaries for configuring and running the Management Agent on the host, and is also referred to as the Core Home.

■ **AGENT_INSTANCE_HOME (or EMSTATE)**  $AGENT_BASE/agent_inst/. Location where the Agent instance files are created, including wrapper scripts to set environment variables, incidents and diagnostic dumps for the Agent, configuration files (principally $AGENT_BASE/agent_inst/sysman/config/emd.properties, the main Agent configuration file), files with target details, and log files. For NFS installations, this directory exists only on remote hosts, and not under $AGENT_BASE.

■ **PLUGIN_HOME**  $AGENT_BASE/plugins/. Location of plug-ins on the Agent side. Each plug-in directory has its own directories for metric metadata, default collections, and scripts specific to the target types the plug-in supports. The plug-ins installed are listed in $AGENT_HOME/sysman/install/plugins.txt. The status of all plug-in installations is contained in $AGENT_HOME/sysman/install/plugins.txt.status.

■ **SBIN_HOME**  $AGENT_BASE/sbin/. This is the setuid executable home that contains important binaries (setuid executables) owned by root, such as nmo, nmb, and nmhs.

The most important files and directories in the Management Agent home directory structure are pictured in Figure 1-2. This will give you a general feel for how the Agent organizes itself to monitor targets.

When an Agent is installed on a host, an entry is added to the /etc/oragchomelist file with the $AGENT_HOME, $OMS_HOME, and $AGENT_INSTANCE_HOME directories. However, an entry for the Agent Home is no longer added to the /etc/oratab file, as with GC 10g/11g Agents.

**Agent Files Uploaded to OMS**
The Agent uploads alerts and monitoring data in XML files to the OMS when any one of the following conditions occurs:

■ The Agent needs to send an alert.

■ The XML files exceed a predefined size.

■ The time since the last upload exceeds a predefined limit (15 minutes by default).

The Agent Upload Manager, also known as the Agent Loader process, is responsible for uploading XML files containing metric data and alerts. The Agent
Chapter 1: Overview of Cloud Control Architecture

FIGURE 1-2. Management Agent directory structure

- AGENT_BASE
  - agent_inst
    - diag
    - config
  - bin
    - nmhs
    - jdk
  - core
    - 12.1.0.1.0
  - plugins
  - nmb
  - sysman
  - admin
  - install
  - plugins.txt
  - plugins.txt
  - status
  - con/fig
  - emd.properties
  -emd
  - log
  - upload
  - metadata
  - default_collection
  - plugins.props
  - scripts

Loader stores these files in the $AGENT_INST/sysman/emd/upload/ upload directory on the target host, which are uploaded to the OMS. The Upload Manager maintains statistics on pending XML files (files on the Agent host not yet uploaded to the OMS) and disables collections based on the number of files in the upload directory (5,000 by default), the aggregate size of these files (50MB by default), and the percentage of free disk space remaining on the upload file system (2 percent by default). The upload interval is 15 minutes by default, and is the time between scheduled uploads of Agent files that contain metric data, not alerts. The upload interval is dynamic because it is based on an Agent property that increases the upload interval by a certain percentage (20 percent by default) for each successive upload failure. All defaults are set in the Agent properties file emd.properties.

Oracle Management Service

The OMS is the middle tier that renders the user interface for the Console. The OMS is deployed on Oracle WebLogic Server (WLS) 11gR1 (10.3.5). As already mentioned, the WLS 10.3.5 installation is bundled with the CC 12c Installer. When running the Installer to create a new EM System, a WLS instance is installed in the subdirectory wlsserver_10.3 under the Middleware home directory (for example, /u01/app/Middleware). The OMS is deployed in a new domain called GCDomain in this WLS instance, and does not use any of the existing built-in domains. Understanding the architecture of an OMS server requires a grasp of each of its elements. WLS 10.3.5 (which contains Oracle HTTP Server) and the OMS J2EE application in a WLS Fusion domain, together comprise what is collectively referred to as the “OMS.”

All OMS components are installed on the same server, regardless of the Cloud Control installation method (via OUI or silent installation). You can deploy multiple Management Services, each on a separate host. However, the OMS on each host must include all OMS components, and must point to the same Repository database for a given Cloud Control environment. Each OMS communicates via thin Java Database Connectivity (JDBC) to the Management Repository. Thin JDBC is a standard Java interface for connecting from Java to relational databases. Also, as mentioned earlier, as of GC 11g, Oracle supports a Management Service installation on a node running an Oracle Real Application Clusters (RAC) 10g or 11g Database that houses a Management Repository.

The OMS directory tree for CC 12c is shown in Figure 1-3.

The Middleware home directory includes subdirectories for the EM instance base directory (gc_inst), JDK 1.6 home (jdk16) home, OMS home (oms), Middleware common home (oracle_common), Middleware WebTier home (Oracle_WT), plug-ins installed on the OMS side (plugins), and WebLogic Server home (wlserver_10.3). The main OMS configuration file, the $OMS_HOME/sysman/config/emoms.properties file used in GC 10g and 11g, is no longer used in CC 12c. Instead, OMS
properties are modified using emctl get/set/delete/list property commands. Once CC is installed, the emctl command lists the various configuration options and their syntax.

The WLS installation provides a WebLogic Server Administration Console, which is a full-feature domain administration site available through a WLS instance-specific URL https://<OMShost>:7101/console or via a link on the GCDomain target home page in the Console. You log in to the WLS Administration Console as the default weblogic user.
As shown in the illustration, the Administration Console connects to the GCDomain, which after installing CC 12c, contains two WLS Managed Servers (and additional servers for each OMS installed):

- **EMGC_ADMINSERVER** The administration server that belongs to the GCDomain.
- **EMGC_OMS1** The OMS application on the first OMS installed. Subsequent OMS installations are named EMGC_OMS2, EMGC_OMS3, and so on.

Both of these two servers will be running when the OMS is started.

**Oracle Management Repository**

The Oracle Management Repository (OMR) is the comprehensive source for all the management information for Cloud Control. It consists of schema definitions, stored procedures, and RDBMS jobs within an Oracle database, all owned by the SYSMAN database user. A particular Cloud Control implementation employs only one Management Repository. You can install multiple Management Services, but each must use this central Repository as their data store.

All CC Console administrators share the same Management Repository information based on the privileges granted to them. Information in the Management Repository includes:

- Configuration details about the managed targets
- Managed target availability information
- Historical metric data and alert information
- Target response time data
- Inventory information on patches and products installed

This information allows administrators to manage their complete application stack (databases, application servers, hosts, and many other target types), model applications, automate tasks, analyze historical status and performance data, perform Application Service Level Management (ASLM), track and make configuration changes, and carry out many other tasks to manage complex IT systems running a combination of Oracle and non-Oracle technologies.

The Management Repository can reside either on the same host as a Management Service or on a dedicated host. You run the CC Installer on the host where you want the OMS to reside, and in the Installer specify the existing database where you want the Installer to create the Management Repository. Oracle supports running the Repository in either a single-instance or Real Application Clusters (RAC) database,
Chapter 1: Overview of Cloud Control Architecture

Release 10gR2, 11gR1, or 11gR2. For specifics on certified database versions, see the Certify page on the My Oracle Support Web site. You must locate the Management Repository in an Enterprise Edition Database. When installing a database for the Repository, you must select the Partitioning Option because Enterprise Manager uses partitioned tables to store management data.

Now let’s examine the Management Repository schema, including the schema owner, tablespaces, and objects within that schema. SYSMAN is both the schema owner and the default Super Administrator account. You cannot remove or rename the SYSMAN account. It is used to

- Perform the initial Cloud Control setup, such as creating privileges and roles, administrator accounts, or notification rules
- Discover new targets
- Create generic jobs to run on all databases or hosts

The Installer for Cloud Control creates three default tablespaces to hold its objects:

- **MGMT_TABLESPACE** Holds all target-related monitoring and metrics data
- **MGMT_ECM_DEPOT_TS** Stores configuration data collected from the monitored targets
- **MGMT_AD4J_TS** Stores diagnostics data related to JVM Diagnostics and Application Dependency Performance (ADP)

As for the schema objects, an open Repository schema is the key to one of Cloud Control’s most important architectural features: extensibility. An open schema means that it is documented so that you can customize the use of Repository data if the standard configuration does not meet your requirements. The Repository tablespaces contain base tables and indexes that begin primarily with MGMT_ and other data types, including over 600 database views. The views, whose names start mostly with “MGMT$”, are particularly handy for mining Repository information that you want to process further. The views are comprehensive so that you can avoid having to directly access the base tables. The inherent advantage to these views is that they insulate custom applications from underlying changes to the base SYSMAN schema due to new releases or patching.

---

6 Configuration management within Cloud Control includes support for Binary Large Objects (BLOBs). Aside from the benefit of logically isolating configuration data in a separate tablespace, it is a best practice for performance and tuning reasons to locate BLOBs in a separate tablespace like this.
Following are the categories of Cloud Control views that provide access to metric, target, and monitoring data stored in the Management Repository:

- Monitoring views
- Inventory views
- Policy Definition views
- Policy Association views
- Policy Violation views
- Management Template views
- Job views
- Application Service Level Management views
- Configuration views
- Oracle Home Patching views
- Linux Patching views
- Security views
- Configuration Management views
- Database Cluster views
- Storage Reporting views

Views provide the basis for passing alerts on to other System Management products, for customizing new plug-ins, reporting problems, and performing historical analysis and data computation.

**Data Flow Between Cloud Control Components**

Now that we’ve reviewed each CC component and its function, let’s look at how these components interact, in order to better understand the CC architectural model. The diagram in Figure 1-4 shows how data flows between components for both Console communications and Agent uploads of metric data and alerts.

Let’s follow the numbers and arrowed lines in this figure to visualize the flow of management data from one component or subcomponent to another. Default ports
are listed in the following explanations. However, an administrator can customize all ports as needed.

1) There are three types of communication for Console requests and Agent uploads:

a) An administrator logs in to the Console at http[s]://<OMShost>:<port>/em through the Oracle HTTP Server (OHS) of the WLS. The default login URL depends on whether the administrator is using a secure or nonsecure connection:

- Secure: https://<OMShost>:7799/em
- Nonsecure: http://<OMShost>:7788/em
b) The Agent uploads metric data and warning alerts to the WLS Oracle HTTP Server URL. The Agent also sends a periodic heartbeat called an Agent ping to its OMS indicating that it is available. The default URL for both types of communication is via a secure or nonsecure connection, as defined by the REPOSITORY_URL property in the Agent configuration file emd.properties:

- Secure: https://<OMHost>:1159/empbs/upload
- Nonsecure: http://<OMHost>:4889/empbs/upload

c) The Agent bypasses the OMS and connects directly to the Repository to report target metadata changes and critical alerts.

2) The administrator request in the Console or Agent upload prompts the OHS to pass the request over HTTP or HTTPS to the OMS application (EMGC_OMS1).

3) The OMS application forwards the Console request or Agent upload to the Repository via thin JDBC. Upon receipt, the OMS renders the content in the Console via the same return path.

4) The OMS communicates to the Agent in several ways:

a) The OMS forwards data directly to the Agent over HTTP(S) to the Agent’s built-in HTTP listener. It listens on the Agent URL http(s): //<OMAhost>:3872/empd/main (https by default), defined by the EMD_URL property located in the Agent configuration file emd.properties. The OMS also submits jobs and other management tasks through this URL.

b) If the OMS-to-Agent communication described in Step 4a is not successful, the OMS checks the status of the target host by pinging the host on which the Agent resides with Internet Control Message Protocol (ICMP) Echo Requests.

c) The OMS sends SQL*Net traffic over TCP ports to target database listeners on Agent hosts, such as when patching database targets through Cloud Control.

Now that you’ve looked more closely at the internal workings of the Cloud Control engine, you’ll have a better understanding of what happens under the hood after CC is installed and operational. This understanding will make it easier to troubleshoot any problems.
Chapter 1: Overview of Cloud Control Architecture

Summary
Here is a synopsis of what was covered in this introductory chapter:

- We began with a discussion of the two principal differences between OEM 12c/11g and GC 10g, which were the use of WebLogic Server versus AS 10g for the OMS middle tier, and the requirement in OEM 12c and 11g that you must preinstall an Oracle database in which the Installer can create the Management Repository.

- Next, we introduced the four main Cloud Control components: the Console, Agent, OMS, and Repository.

- We ended this chapter by examining the interaction between Cloud Control components. You saw how these components work together to deliver Console requests and metric data from the Agent on a target host through the OMS to the Repository, how the OMS in turn delivers data back to the Agent, and how Cloud Control performs self-monitoring by using one component to check the status of another.

Now that you understand the basic architecture and concepts of Cloud Control, you are ready to proceed to Chapter 2 to perform CC preinstallation steps. In Chapter 3 you will create the database for the Management Repository, and then, in Chapter 4, you will install Cloud Control.