RAMPED ORDERING FOR CLOUD SERVICES

Applicant: ORACLE INTERNATIONAL CORPORATION, Redwood Shores, CA (US)

Inventors: Ramkrishna Chatterjee, Nashua, NH (US); Ramesh Vasudevan, Los Altos, CA (US); Pradeep Seetharam, Fremont, CA (US); Gopalan Arun, Saratoga, CA (US)

Assignee: Oracle International Corporation, Redwood Shores, CA (US)

Appl. No.: 13/841,768

Filed: Mar. 15, 2013


Publication Classification

Int. Cl.
G06Q 30/06
G06Q 30/0635

U.S. Cl.
P168 708/26.81

ABSTRACT

Provisioning, managing and tracking of services provided by a cloud infrastructure system are described. A subscription request from a customer for one or more services provided by the cloud infrastructure system is received. The request includes a specification of services with different service levels for different time periods. A phased deployment of a customer’s subscription order based on the service levels and time periods specified by the customer in the subscription order is then determined.
PROCESS SUBSCRIPTION ORDER 302

IDENTIFY BUSINESS PROCESS ASSOCIATED WITH ORDER 304

EXECUTE BUSINESS PROCESS ASSOCIATED WITH ORDER 306

SEND NOTIFICATION REGARDING STATUS OF PROVISIONED ORDER 308

FIG. 3A
FIG. 4
FIG. 6
RECEIVE BUSINESS PROCESS ASSOCIATED WITH ORDER 702

TRANSLATE BUSINESS PROCESS INTO SERIES OF TASKS 704

PROVISION RESOURCES BASED ON SERIES OF TASKS 706

FIG. 7A
FIG. 7D
FIG. 8

ORDER PROCESSING MODULE 310

ORDER SCHEDULING 802

PROVISIONING ORDER GENERATOR 800

HEURISTICS MODULE 804

INPUT SOURCES 322

SDI COMPONENT 206

BUSINESS PROCESS IDENTIFIER 312

BUSINESS PROCESS EXECUTOR 316
RECEIVE A SUBSCRIPTION ORDER 902

IDENTIFY A REQUEST FOR A SERVICE AT A FIRST SERVICE LEVEL FOR A FIRST TIME PERIOD 904

SCHEDULE BASE ORDER FOR EXECUTION AT THE FIRST TIME PERIOD 906

IDENTIFY REQUESTS FOR ADDITIONAL SERVICE LEVELS FOR THE SERVICE FOR FUTURE TIME PERIODS 908

SCHEDULE THE ADDITIONAL ORDERS FOR EXECUTION AT THE FUTURE TIME PERIODS 910

FIG. 9
RAMPED ORDERING FOR CLOUD SERVICES

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] The present application is a non-provisional of and claims the benefit and priority under 35 U.S.C. 119(e) of the following applications, the entire contents of which are incorporated herein by reference for all purposes:

[0002] (1) U.S. Provisional Application No. 61/698,413, filed Sep. 7, 2012, entitled TENANT AUTOMATION SYSTEM;

[0003] (2) U.S. Provisional Application No. 61/698,459, filed Sep. 7, 2012, entitled SERVICE DEVELOPMENT INFRASTRUCTURE; and

[0004] (3) U.S. Provisional Application No. 61/785,299, filed Mar. 14, 2013, entitled CLOUD INFRASTRUCTURE.

BACKGROUND

[0005] The present disclosure relates to computer systems and software, and more particularly to techniques for facilitating and automating the provision of services in a cloud environment.

[0006] Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurations and computational resources (e.g., networks, servers, storage, applications, and services). The services provided or accessed through the cloud (or network) are referred to as cloud services. There is a lot of processing that needs to be performed by a cloud service provider to make cloud services available to a subscribing customer. Due to its complexity, much of this processing is still done manually. For example, provisioning resources for providing such cloud services can be a very labor-intensive process.

SUMMARY

[0007] Certain embodiments of the present invention provide techniques for automating the provisioning, managing and tracking of services provided by a cloud infrastructure system. In one embodiment, the cloud infrastructure system stores subscription order information related to one or more services subscribed to by a customer in the cloud infrastructure system.

[0008] In one embodiment, upon receiving a subscription request from a customer for one or more services, the cloud infrastructure system includes techniques for providing the customer with the ability to request services in a subscription order with different service levels for different time periods. In this manner, a customer is provided with the ability to subscribe to a multi-year (or other time period) plan, with changing service levels (increasing/decreasing) from one year to the next. The manner in which the service levels are defined maybe service-specific and may vary from one service to another. In one embodiment, the service levels relate to the amount of resources assigned to one or more services in the subscription order. In one embodiment, techniques are provided for performing a phased deployment of a customer’s subscription order based on the service levels and time periods specified by the customer in the subscription order.

[0009] Some embodiments relate to a method for receiving a subscription order requesting a service offered by the cloud infrastructure system. The method includes identifying a request for a first service level to be provided for the service for a first time period, based on the subscription order. The method then includes a base order to be scheduled for execution for the first time period. The base order corresponds to the service at the first service level for the first time period. Then, the method includes identifying, based upon the subscription order, a request for additional service levels to be provided for the service for future time periods. In one embodiment, the future time periods occur after the first time period. The additional orders are then to be scheduled for execution for the future time periods. The additional orders correspond to the additional service levels of the service to be provided over the future time periods.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Illustrative embodiments of the present invention are described in detail below with reference to the following drawing figures:

[0011] FIG. 1A is a logical view of a cloud infrastructure system according to one embodiment of the present invention.

[0012] FIG. 1B is a simplified block diagram of a hardware/software stack that may be used to implement a cloud infrastructure system according to an embodiment of the present invention.

[0013] FIG. 2 is a simplified block diagram of a system environment for implementing the cloud infrastructure system shown in FIG. 1A.

[0014] FIG. 3A depicts a simplified flowchart depicting processing that may be performed by the TAS module in the cloud infrastructure system, in accordance with an embodiment of the present invention.

[0015] FIG. 3B depicts a simplified high level diagram of one or more sub-modules in the TAS module in the cloud infrastructure system, in accordance with an embodiment of the present invention.

[0016] FIG. 4 depicts an exemplary distributed deployment of the TAS component, according to an embodiment of the present invention.

[0017] FIG. 5 is a simplified block diagram illustrating the interactions of the SDI module with one or more modules in the cloud infrastructure system, in accordance with an embodiment of the present invention.

[0018] FIG. 6 depicts a simplified high level diagram of sub-modules of the SDI module according to an embodiment of the present invention.

[0019] FIG. 7A depicts a simplified flowchart depicting processing that may be performed by the SDI component in the cloud infrastructure system, in accordance with an embodiment of the present invention.

[0020] FIG. 7B depicts a simplified block diagram showing the high-level architecture of a Nuviaq system 710 and its relationships with other cloud infrastructure components according to an embodiment of the present invention.

[0021] FIG. 7C depicts an example sequence diagram illustrating steps of a provisioning process using a Nuviaq system according to an embodiment of the present invention.

[0022] FIG. 7D depicts an example sequence diagram illustrating steps of a deployment process using a Nuviaq system according to an embodiment of the present invention.

[0023] FIG. 7E depicts an example of database instances provisioned for a database service according to an embodiment of the present invention.
FIG. 8 is a high level block diagram illustrating modules in the cloud infrastructure system for performing a phased deployment of a customer’s subscription order, in accordance with one embodiment of the present invention.

FIG. 9 depicts a simplified flowchart depicting processing that may be performed by the order processing module in the cloud infrastructure system, in accordance with an embodiment of the present invention.

FIG. 10 is a simplified block diagram of a computing system that may be used in accordance with embodiments of the present invention.

DETAILED DESCRIPTION

In the following description, for the purposes of explanation, specific details are set forth in order to provide a thorough understanding of embodiments of the invention. However, it will be apparent that various embodiments may be practiced without these specific details. The figures and description are not intended to be restrictive.

Certain embodiments of the present invention provide techniques for automating the provisioning, managing and tracking of services provided by a cloud infrastructure system.

In certain embodiments, a cloud infrastructure system may include a suite of applications, middleware and database service offerings that are delivered to a customer in a self-service, subscription-based, elastically scalable, reliable, highly available, and secure manner. An example of such a cloud infrastructure system is the Oracle Public Cloud provided by the present assignee.

A cloud infrastructure system may provide many capabilities including, but not limited to, provisioning, managing and tracking a customer’s subscription for services and resources in the cloud infrastructure system, providing predictable operating expenses to customers utilizing the services in the cloud infrastructure system, providing robust identity domain separation and protection of a customer’s data in the cloud infrastructure system, providing customers with a transparent architecture and control of the design of the cloud infrastructure system, providing customers secured data protection and compliance with data privacy standards and regulations, providing customers with an integrated development experience for building and deploying services in the cloud infrastructure system and providing customers with a seamless integration between business software, middleware, database and infrastructure services in the cloud infrastructure system.

In certain embodiments, services provided by the cloud infrastructure system may include a host of services that are made available to users of the cloud infrastructure system on demand such as online data storage and backup solutions, Web-based e-mail services, hosted office suites and document collaboration services, database processing, managed technical support services and the like. Services provided by the cloud infrastructure system can dynamically scale to meet the needs of its users. A specific instantiation of a service provided by cloud infrastructure system is referred to herein as a service instance. In general, any service made available to a user via a communication network such as the Internet from a cloud service provider’s system is referred to as a cloud service. Typically, in a public cloud environment, servers and systems that make up the cloud service provider’s system are different from the customer’s own on-premises servers and systems. For example, a cloud service provider’s system may host an application and a user may, via a communication network such as the Internet, on demand, order and use the application.

A service in a computer network cloud infrastructure includes protected computer network access to storage, a hosted database, a hosted web server, a software application, or other service provided by a cloud vendor to a user, or as otherwise known in the art. For example, a service can include password-protected access to remote storage on the cloud through the Internet. As another example, a service can include a web service-based hosted relational database and script-language middleware engine for private use by a networked developer. As another example, a service can include access to an email software application hosted on a cloud vendor’s web site.

FIG. 1A is a logical view of a cloud infrastructure system according to one embodiment of the present invention. Cloud infrastructure system 100 may provide a variety of services via a cloud or networked environment. These services may include one or more services provided under Software as a Service (SaaS) category, Platform as a Service (PaaS) category, Infrastructure as a Service (IaaS) category, or other categories of services including hybrid services. A customer, via a subscription order, may order one or more services provided by cloud infrastructure system 100. Cloud infrastructure system 100 then performs processing to provide the services in the customer’s subscription order.

Cloud infrastructure system 100 may provide the cloud services via different deployment models. For example, services may be provided under a public cloud model where cloud infrastructure system 100 is owned by an organization selling cloud services (e.g., owned by Oracle) and the services are made available to the general public or different industry enterprises. As another example, services may be provided under a private cloud model where cloud infrastructure system 100 is operated solely for a single organization and may provide services for one or more entities within the organization. The cloud services may also be provided under a community cloud model where cloud infrastructure system 100 and the services provided by system 100 are shared by several organizations in a related community. The cloud services may also be provided under a hybrid cloud model, which is a combination of two or more different models.

As shown in FIG. 1A, cloud infrastructure system 100 may comprise multiple components, which working in conjunction, enable provision of services provided by cloud infrastructure system 100. In the embodiment illustrated in FIG. 1A, cloud infrastructure system 100 includes a SaaS platform 102, a PaaS platform 104, an IaaS platform 110, infrastructure resources, and cloud management functionality. These components may be implemented in hardware, or software, or combinations thereof.

SaaS platform 102 is configured to provide cloud services that fall under the SaaS category. For example, SaaS platform 102 may provide capabilities to build and deliver a suite of on-demand applications on an integrated development and deployment platform. SaaS platform 102 may manage and control the underlying software and infrastructure for providing the SaaS services. By utilizing the services provided by SaaS platform 102, customers can utilize applications executing on cloud infrastructure system 100. Customers can acquire the application services without the need for customers to purchase separate licenses and support.
Various different SaaS services may be provided. Examples include without limitation services that provide solutions for sales performance management, enterprise integration and business flexibility for large organizations, and the like. In one embodiment, the SaaS services may include Customer Relationship Management (CRM) services 110 (e.g., Fusion CRM services provided by the Oracle cloud), Human Capital Management (HCM)/Talent Management services 112, and the like. CRM services 110 may include services directed to managing and optimization of a sales activity cycle to a customer, and others. HCM/Talent services 112 may include services directed to providing global workforce lifecycle management and talent management services to a customer.

Various different PaaS services may be provided by PaaS platform 104 in a standardized, shared and elastically scalable application development and deployment platform. Examples of PaaS services may include without limitation services that enable organizations (such as Oracle) to consolidate existing applications on a shared, common architecture, as well as the ability to build new applications that leverage the shared services provided by the platform. PaaS platform 104 may manage and control the underlying software and infrastructure for providing the PaaS services. Customers can acquire the PaaS services provided by cloud infrastructure system 100 without the need for customers to purchase separate licenses and support. Examples of PaaS services include without limitation Oracle Java Cloud Service (JCS), Oracle Database Cloud Service (DBCS), and others.

By utilizing the services provided by PaaS platform 104, customers can utilize programming languages and tools supported by cloud infrastructure system 100 and also control the deployed services. In some embodiments, PaaS services provided by the cloud infrastructure system 100 may include database cloud services 114, middleware cloud services (e.g., Oracle Fusion Middleware services) 116 and Java cloud services 117. In one embodiment, database cloud services 114 may support shared service deployment models that enable organizations to pool database resources and offer customers a database-as-a-service in the form of a database cloud, middleware cloud services 116 provides a platform for customers to develop and deploy various business applications and Java cloud services 117 provides a platform for customers to deploy Java applications, in the cloud infrastructure system 100. The components in SaaS platform 102 and PaaS platform 104 illustrated in FIG. 1A are meant for illustrative purposes only and are not intended to limit the scope of embodiments of the present invention. In alternate embodiments, SaaS platform 102 and PaaS platform 104 may include additional components for providing additional services to the customers of cloud infrastructure system 100.

Various different IaaS services may be provided by IaaS platform 110. The IaaS services facilitate the management and control of the underlying computing resources such as storage, networks, and other fundamental computing resources for customers utilizing services provided by the SaaS platform and the PaaS platform.

In certain embodiments, cloud infrastructure system 100 includes infrastructure resources 106 for providing the resources used to provide various services to customers of the cloud infrastructure system 100. In one embodiment, infrastructure resources 106 includes pre-integrated and optimized combinations of hardware such as servers, storage and networking resources to execute the services provided by the PaaS platform and the SaaS platform.

In certain embodiments, cloud management functionality 108 provides comprehensive management of cloud services (e.g., SaaS, PaaS, IaaS services) in the cloud infrastructure system 100. In one embodiment, cloud management functionality 108 includes capabilities for provisioning, managing and tracking a customer's subscription received by the cloud infrastructure system 100, and the like.

FIG. 1B is a simplified block diagram of a hardware/software stack that may be used to implement cloud infrastructure system 100 according to an embodiment of the present invention. It should be appreciated that implementation depicted in FIG. 1B may have other components than those depicted in FIG. 1B. Further, the embodiment shown in FIG. 1B is only one example of a cloud infrastructure system that may incorporate an embodiment of the invention. In some other embodiments, cloud infrastructure system 100 may have more or fewer components than shown in FIG. 1B, may combine two or more components, or may have a different configuration or arrangement of components. In certain embodiments, the hardware and software components are stacked so as to provide vertical integration that provides optimal performance.

Various types of users may interact with cloud infrastructure system 100. These users may include, for example, end users 150 that can interact with cloud infrastructure system 100 using various client devices such as desktops, mobile devices, tablets, and the like. The users may also include developers/programmers 152 who may interact with cloud infrastructure system 100 using command line interfaces (CLIs), application programming interfaces (APIs), through various integrated development environments (IDEs), and via other applications. User may also include operations personnel 154. These may include personnel of the cloud service provider or personnel of other users.

Application services layer 156 identifies various cloud services that may be offered by cloud infrastructure system 100. These services may be mapped to or associated with respective software components 160 (e.g., Oracle WebLogic server for providing Java services, oracle database for providing database services, and the like) via a service integration and linkages layer 158.

In certain embodiments, a number of internal services 162 may be provided that are shared by different components or modules of cloud infrastructure system 100 and by the services provided by cloud infrastructure system 100. These internal shared services may include, without limitation, a security and identity service, an integration service, an enterprise repository service, an enterprise manager service, a virus scanning and white list service, a high availability, backup and recovery service, service for enabling cloud support in IDEs, an email service, a notification service, a file transfer service, and the like.

Runtime infrastructure layer 164 represents the hardware layer on which the various other layers and components are built. In certain embodiments, runtime infrastructure layer 164 may comprise one Oracle's Exadata machines for providing storage, processing, and networking resources. An Exadata machine may be composed of various database servers, storage Servers, networking resources, and other components for hosting cloud-services related software layers. In certain embodiments, the Exadata machines may be designed to work with Oracle Exalogic, which is an engi-
neered system providing an assemblage of storage, compute, network, and software resources. The combination of Exta
data and Exalogic provides a complete hardware and software engineered solution that delivers high-performance, highly available, scalable, secure, and a managed platform for providing cloud services.

[0048] FIG. 2 is a simplified block diagram of a system environment for implementing the cloud infrastructure sys
tem shown in FIG. 1A according to an embodiment of the present invention. In the illustrated embodiment, system environment 230 includes one or more client computing devices 224, 226 and 228 that may be used by users to interact with cloud infrastructure system 100. A client device may be configured to operate a client application such as a web browser, a proprietary client application (e.g., Oracle Forms), or some other application, which may be used by a user of the client device to interact with cloud infrastructure system 100 to utilize services provided by cloud infrastructure system 100.

[0049] It should be appreciated that cloud infrastructure system 100 depicted in FIG. 2 may have other components than those depicted in FIG. 2. Further, the embodiment shown in FIG. 2 is only one example of a cloud infrastructure system that may incorporate an embodiment of the invention. In some other embodiments, cloud infrastructure system 100 may have more or fewer components than shown in FIG. 2, may combine two or more components, or may have a different configuration or arrangement of components.

[0050] Client computing devices 224, 226 and 228 may be a general purpose personal computers (including, by way of example, personal computers and/or laptop computers running various versions of Microsoft Windows and/or Apple Macintosh operating systems), cell phones or PDAs (running software such as Microsoft Windows Mobile and being Internet, e-mail, SMS, Blackberry, or other communication protocol enabled), workstation computers running any of a variety of commercially available UNIX or UNIX-like operating systems (including without limitation the variety of GNU/Linux operating systems), or any other computing device. For example, client computing devices 224, 226 and 228 may be any other electronic device, such as a thin-client computer, Internet-enabled gaming system, and/or personal messaging device, capable of communicating over a network (e.g., network 232 described below). Although exemplary system environment 230 is shown with three client computing devices, any number of client computing devices may be supported. Other devices such as devices with sensors, etc. may interact with cloud infrastructure system 100.

[0051] A network 232 may facilitate communications and exchange of data between clients 224, 226 and 228 and cloud infrastructure system 100. Network 232 may be any type of network familiar to those skilled in the art that can support data communications using any of a variety of commercially available protocols, including without limitation TCP/IP, SNA, IPX, AppleTalk, and the like. Merely by way of example, network 232 can be a local area network (LAN) such as an Ethernet network, a Token-Ring network and/or the like, a wide-area network, a virtual network, including without limitation a virtual private network (VPN), the Internet, an intranet, an extranet, a public switched telephone network (PSTN), an infra-red network, a wireless network (e.g., a network operating under any of the IEEE 802.1X suite of protocols, the Bluetooth protocol known in the art, and/or any other wireless protocol), and/or any combination of these and/or other networks.

[0052] Cloud infrastructure system 100 may comprise one or more computers and/or servers which may be general purpose computers, specialized server computers (including, by way of example, PC servers, UNIX servers, mid-range servers, mainframe computers, rack-mounted servers, etc.), server farms, server clusters, or any other appropriate arrangement and/or combination. The computing devices that make up cloud infrastructure system 100 may run any of the various services associated with operating systems or a variety of additional server applications and/or mid-tier applications, including HTTP servers, FTP servers, CGI servers, Java servers, database servers, and the like. Exemplary database servers include without limitation those commercially available from Oracle, Microsoft, Sybase, IBM and the like.

[0053] In various embodiments, cloud infrastructure system 100 may be adapted to automatically provision, manage and track a customer's subscription to services offered by cloud infrastructure system 100. In one embodiment, as depicted in FIG. 2, the components in cloud infrastructure system 100 include an Identity Management (IDM) module 200, a services module 202, a Tenant Automation System (TAS) module 204, a Service Deployment Infrastructure (SDI) module 206, an Enterprise Manager (EM) module 208, one or more front-end web interfaces such as a store user interface (UI) 210, a user interface (UI) 212, and a support user interface (UI) 216, an order management module 214, sales personnel 218, operator personnel 220 and an order database 224. These modules may include or be provided using one or more computers and/or servers which may be general purpose computers, specialized server computers, server farms, server clusters, or any other appropriate arrangement and/or combination. In one embodiment, one or more of these modules can be provided by cloud management functionality 108 or IaaS platform 110 in cloud infrastructure system 100. The various modules of the cloud infrastructure system 100 depicted in FIG. 2 are meant for illustrative purposes only and are not intended to limit the scope of embodiments of the present invention. Alternative embodiments may include more or fewer modules than those shown in FIG. 2.

[0054] In an exemplary operation, at (1) a customer using a client device such as client device 224 or 226 may interact with cloud infrastructure system 100 by browsing the various services provided by cloud infrastructure system 100 and placing an order for a subscription for one or more services offered by cloud infrastructure system 100. In certain embodiments, the customer may access store UI 210 or cloud UI 212 and place a subscription order via these user interfaces.

[0055] The order information received by cloud infrastructure system 100 in response to the customer placing an order may include information identifying the customer and one or more services offered by the cloud infrastructure system 100 that the customer intends to subscribe to. A single order may include orders for multiple services. For instance, a customer may login to cloud UI 212 and request a subscription for a CRM service and a Java cloud service in the same order.

[0056] Additionally, the order may also include one or more service levels for the requested services. As used herein, and as will be discussed in greater detail below, a service level for a service determines the amount of resources to be allocated for providing the requested service in the context of the
subscription, such as the amount of storage, amount of computing resources, data transfer facilities, and the like. For example, a basic service level may provide a minimum level of storage, data transmission, or number of users, and higher service levels may include additional resources.

[0057] In addition, in some instances, the order information received by cloud infrastructure system 100 may include information indicative of a customer level, and the time period during which the service is desired. The customer level specifies the priority of the customer making the subscription request. In one example, the priority may be determined based on the quality of service that the cloud infrastructure system 100 guarantees or promises the customer as specified by a Service Level Agreement (SLA) agreed to between the customer and the provider of the cloud services. In one example, the different customer levels include a basic level, a silver level and a gold level. The time period for a service may specify the start date and time for the service and the time period for which the service is desired (e.g., a service end date and time may be specified).

[0058] In one embodiment, a customer may request a new subscription via store UI 210 or request for a trial subscription via cloud UI 212. In certain embodiments, store UI 210 may represent the service provider’s e-commerce store front (e.g., www.oracle.com/store for Oracle Cloud services). Cloud UI 212 may represent a business interface for the service provider. Consumer can explore available services and sign up for interested services through cloud UI 212. Cloud UI 212 captures user input necessary for ordering trial subscriptions provided by cloud infrastructure system 100. Cloud UI 212 may also be used to view account features and configure the runtime environment located within cloud infrastructure system 100. In addition to placing an order for a new subscription, store UI 210 may also enable the customer to perform other subscription-related tasks such as changing the service level of a subscription, extending the term of the subscription, increasing the service level of a subscription, terminating an existing subscription, and the like.

[0059] After an order has been placed per (1), at (2), the order information that is received via either store UI 210 or cloud UI 212 is stored in order database 224, which can be one of several databases operated by cloud infrastructure system 100 and utilized in conjunction with other system elements. While order database 224 is shown logically as a single database in FIG. 2, in actual implementation, this may comprise one or more databases.

[0060] At (3), the order is forwarded to order management module 214. Order management module 214 is configured to perform billing and accounting related functions related to the order as such as verifying the order and upon verification, booking the order. In certain embodiments, order management module 214 may include a contract management module and an install base module. The contract management module may store contract information associated with the customer’s subscription order such as the customer’s service level agreement (SLA) with cloud infrastructure system 100. The install base module may include detailed descriptions of the services in the customer’s subscription order. In addition to order information, the install base module may track installation details related to the services, product status and support service history related to the services. As a customer orders new services or upgrades existing ones, the install base module may automatically add new order information.

[0061] At (4), information regarding the order is communicated to TAS module 204. In one embodiment, TAS module 204 utilizes the order information to orchestrate the provisioning of services and resources for the order placed by the customer. At (5), TAS component 204 orchestrates the provisioning of resources to support the subscribed services using the services of SDI module 206. At (6) TAS module 204 provides information related to the provisioned order received from SDI module 206 to services module 202. In some embodiments, at (7), SDI module 206 may also use services provided by services module 202 to allocate and configure the resources needed to fulfill the customer’s subscription order.

[0062] At (8), services module 202 sends a notification to the customers on client devices 224, 226 and 228 regarding the status of the order.

[0063] In certain embodiments, TAS module 204 functions as an orchestration component that manages business processes associated with each order and applies business logic to determine whether an order should proceed to provisioning. In one embodiment, upon receiving an order for a new subscription, TAS module 204 sends a request to SDI module 206 to allocate resources and configure those resources needed to fulfill the subscription order. SDI module 206 enables the allocation of resources for the services ordered by the customer. SDI module 206 provides a level of abstraction between the cloud services provided by cloud infrastructure system 100 and the physical implementation layer that is used to provision the resources for providing the requested services. TAS module 204 may thus be isolated from implementation details such as whether or not services and resources are actually provisioned on the fly or pre-provisioned and only allocated/assigned upon request.

[0064] In certain embodiments, a user may use store UI 210 to directly interact with order management module 214 to perform billing and accounting related functions such as verifying the order and upon verification, booking the order. In some embodiments, instead of a customer placing an order, at (9), the order may instead be placed by sales personnel 218 on behalf of the customer such as a customer’s service representative or sales representative. Sales personnel 218 may directly interact with order management module 214 via a user interface (not shown in FIG. 2) provided by order management module 214 for placing orders or for providing quotes for the customer. This, for example, may be done for large customers where the order may be placed by the customer’s sales representative through order management module 214. The sales representative may set up the subscription on behalf of the customer.

[0065] EM module 208 is configured to monitor activities related to managing and tracking a customer’s subscription in cloud infrastructure system 100. EM module 208 collects usage statistics for the services in the subscription order such as the amount of storage used, the amount of data transferred, the number of users, and the amount of system uptime and system down time. At (10), a host operator personnel 220, who may be an employee of a provider of cloud infrastructure system 100, may interact with EM module 208 via an enterprise manager user interface (not shown in FIG. 2) to manage systems and resources on which services are provisioned within cloud infrastructure system 100.

[0066] Identity management (IDM) module 200 is configured to provide identity services such as access management and authorization services in cloud infrastructure system 100.
In one embodiment, IDM module 200 controls information about customers who wish to utilize the services provided by cloud infrastructure system 100. Such information can include information that authenticates the identities of such customers and information that describes which actions those customers are authorized to perform relative to various system resources (e.g., files, directories, applications, communication ports, memory segments, etc.) IDM module 200 can also include the management of descriptive information about each customer and about how and by whom that descriptive information can be accessed and modified.

[0067] In one embodiment, information managed by the identity management module 200 can be partitioned to create separate identity domains. Information belonging to a particular identity domain can be isolated from all other identity domains. Also, an identity domain can be shared by multiple separate tenants. Each such tenant can be a customer subscribing to services in the cloud infrastructure system 100. In some embodiments, a customer can have one or many identity domains, and each identity domain may be associated with one or more subscriptions, each subscription having one or many services. For example, a single customer can represent a large entity and identity domains may be created for divisions/departments within this large entity. EM module 208 and IDM module 200 may in turn interact with order management module 214 at (11) and (12) respectively to manage and track the customer's subscriptions in cloud infrastructure system 100.

[0068] In one embodiment, at (13), support services may also be provided to the customer via a support UI 216. In one embodiment, support UI 216 enables support personnel to interact with order management module 214 via a support backend system to perform support services at (14). Support personnel in the cloud infrastructure system 100 as well as customers can submit bug reports and check the status of these reports via support UI 216.

[0069] Other interfaces, not shown in FIG. 2 may also be provided by cloud infrastructure system 100. For example, an identity domain administrator may use a user interface to IDM module 200 to configure domain and user identities. In addition, customers may log into a separate interface for each service they wish to utilize. In certain embodiments, a customer who wishes to subscribe to one or more services offered by cloud infrastructure system 100 may also be assigned various roles and responsibilities. In one embodiment, the different roles and responsibilities that may be assigned for a customer may include that of a buyer, an account administrator, a service administrator, an identity domain administrator or a user who utilizes the services and resources offered by cloud infrastructure system 100. The different roles and responsibilities are described more fully in FIG. 4 below.

[0070] FIG. 3A depicts a simplified flowchart 300 depicting processing that may be performed by the TAS module in the cloud infrastructure system, in accordance with an embodiment of the present invention. The processing depicted in FIG. 3A may be implemented in software (e.g., code, instructions, program) executed by one or more processors, hardware, or combinations thereof. The software may be stored in memory (e.g., on a memory device, on a non-transitory computer-readable storage medium). The particular series of processing steps depicted in FIG. 3A is not intended to be limiting. Other sequences of steps may also be performed according to alternative embodiments. For example, alternative embodiments of the present invention may perform the steps outlined above in a different order. Moreover, the individual steps illustrated in FIG. 3A may include multiple sub-steps that may be performed in various sequences as appropriate to the individual step. Furthermore, additional steps may be added or removed depending on the particular applications. One of ordinary skill in the art would recognize many variations, modifications, and alternatives. In one embodiment, the processing depicted in FIG. 3A may be performed by one or more components in TAS component 204 as will be described in detail in FIG. 3B.

[0071] At 302, a customer's subscription order is processed. The processing may include validating the order, in one example. Validating the order includes ensuring that the customer has paid for the subscription and ensuring that the customer does not already have subscriptions with the same name or that the customer is not attempting to create multiple subscriptions of the same type in the same identity domain for subscription types for which this is disallowed (such as, in the case of a CRM service). Processing may also include tracking the status of an order for each order that is being processed by cloud infrastructure system 100.

[0072] At 304, a business process associated with the order is identified. In some instances, multiple business processes may be identified for an order. Each business process identifies a series of steps for processing various aspects of the order. As an example, a first business process may identify one or more steps related to provisioning physical resources for the order, a second business process may identify one or more steps related to creating an identity domain along with customer identities for the order, a third business process may identify one or more steps related to performing back office functions such as creating a customer record for the user, performing accounting functions related to the order, and the like. In certain embodiments, different business processes may also be identified for processing different services in an order. For example, different business process may be identified to process a CRM service and a database service.

[0073] At 306, the business process identified for the order in 304 is executed. Executing the business process associated with the order may include orchestrating the series of steps associated with the business process identified in step 304. For example, executing a business process related to provisioning physical resources for the order may include sending a request to SDI module 206 to allocate resources and configure those resources needed to fulfill the subscription order.

[0074] At 308, a notification is sent to the customer regarding the status of the provisioned order. Additional description related to performing steps 302, 304, 306 and 308 is provided in detail in FIG. 3B.

[0075] FIG. 3B depicts a simplified high level diagram of one or more sub-modules in the TAS module in the cloud infrastructure system, in accordance with an embodiment of the present invention. In one embodiment, the modules depicted in FIG. 3B perform the processing described in steps 302-308 discussed in FIG. 3A. In the illustrated embodiment, TAS module 204 comprises an order processing module 310, a business process identifier 312, a business process executor 316, an overage framework 322, a workflow identification module 324, and a bundled subscription generator module 326. These modules may be implemented in hardware, software, or combinations thereof. The various modules of the TAS module depicted in FIG. 3B are meant for illustrative purposes only and are not intended to limit the scope of
embodiments of the present invention. Alternative embodiments may include more or fewer modules than those shown in FIG. 3B.

[0076] In one embodiment, order processing module 310 receives an order from a customer from one or more input sources 321. For example, order processing module 310 may directly receive an order via cloud UI 212 or store UI 210, in one embodiment. Alternatively, order processing module 310 may receive an order from order management module 214 or order database 224. Order processing module 310 then processes the order. In certain embodiments, processing the order includes generating a customer record which includes information about the order such as a service type, a service level, a customer level, the type of resources, the amount of the resources to be allocated to the service instance and a time period during which the service is desired. As part of the processing, order processing module 310 also determines whether the order is a valid order. This includes ensuring that the customer does not already have subscriptions with the same name or that the customer is not attempting to create multiple subscriptions of the same type in the same identity domain for subscription types where this is disallowed (such as, in the case of a fusion CRM service).

[0077] Order processing module 310 may also perform additional processing on the order. Processing may include tracking the status of an order for each order that is being processed by cloud infrastructure system 100. In one embodiment, order processing module 310 may process each order to identify a number of states pertaining to the order. In one example, the different states of an order may be an initialized state, a provisioned state, an active state, and an administration required state, an error state, and the like. An initialized state refers to the state of a new order; a provisioned state refers to the state of an order once the services and resources for the order have been provisioned. An order is in an active state when the order has been processed by TAS module 204 and a notification to that effect has been delivered to the customer. An order is in an administration required state when intervention by an administrator is needed to resolve the issue. The order is in an error state when the order cannot be processed. In addition to maintaining the order progress status, order processing module 310 also maintains detailed information about any failures encountered during process execution. In other embodiments, and as will be discussed in detail below, the additional processing performed by order processing module 310 may also include changing the service level for a service in the subscription, changing the services included in the subscription, extending the time period of the subscription, and canceling the subscription or specifying different service levels for different time periods in the subscription.

[0078] After an order has been processed by order processing module 310, business logic is applied to determine whether the order should proceed to provisioning. In one embodiment, as part of orchestrating the order, business process identifier 312 receives the processed order from order processing module 310 and applies business logic to identify a particular business process to use for the order being processed. In one embodiment, business process identifier 312 may utilize information stored in a service catalog 314 to determine the particular business process to be used for the order. In another embodiment, and as discussed above, different business processes may be defined for different types of services, or combinations of services such as a CRM service or a database service. In one embodiment, service catalog 314 may store information mapping an order to a particular type of business process. Business process identifier 312 may use this information to identify a specific business process for the order being processed. Once a business process has been identified, business process identifier 312 communicates the particular business process to be executed to business process executor 316. Business process executor 316 then executes steps of the identified business process by operating in conjunction with one or more modules in the cloud infrastructure system 100. In some embodiments, business process executor 316 acts as an orchestrator for performing the steps associated with a business process. For example, the business process executor may interact with order processing module 310 to execute steps in a business process that identifies workflows related to the order, determines the average overages in the order or identifies service components related to the order.

[0080] In one example, business process executor 316 interacts with SDI module 206 to execute steps in a business process for allocating and provisioning resources for services requested in the subscription order. In this example, for each step in the service process, business process executor 316 may send a request to SDI component 206 to allocate resources and configure resources needed to fulfill the particular step. SDI component 206 is responsible for the actual allocation of the resources. Once all the steps of the business processes of an order have been executed, business process executor 316 may send a notification to the customer of the processed order by utilizing the services of service component 202. The notification may include sending an event notification to the customer with details of the processed order. The event notification may also include deployment information related to the order to enable the customer to access the subscribed services.

[0081] In certain embodiments, TAS module 204 may provide one or more TAS Application Programming Interfaces (APIs) 318 that enable TAS module 204 to interact with other modules in cloud infrastructure system 100 and for other modules to interact with TAS module 204. For example, the TAS APIs may include a system provisioning API that interacts with SDI module 206 via an asynchronous Simple Object Access Protocol (SOAP) based web services call to provision resources for the customer's subscription order. In one embodiment, TAS module 204 may also utilize the system provisioning API to accomplish system and service instance creation and deletion, switch a service instance to an increased service level, and associate service instances. An example of this is the association of a Java service instance to a fusion applications service instance to allow secure web service communications. The TAS APIs may also include a notification API that interacts with the services module 202 to notify the customer of a processed order. In certain embodiments, the TAS module 204 also periodically propagates subscription information, outages, and notifications (e.g. planned downtime) to services component 202.

[0082] In certain embodiments, TAS module 204 periodically receives usage statistics for each of the provisioned services such as the amount of storage used, the amount data transferred, the number of users, and the amount of system up time and system down time from EM module 208. Overage framework 322 utilizes the usage statistics to determine
whether over use of a service has occurred, and if so, to determine how much to bill for the overage, and provides this information to order management module 214.

In certain embodiments, TAS module 204 includes an order workflow identification module 324 that is configured to identify one or more workflows associated with processing a customer’s subscription order. In certain embodiments, TAS module 204 may include a subscription order generation framework 326 for generating subscription orders for a customer when the customer places a subscription order for one or more services offered by the cloud infrastructure system 100. In one embodiment, a subscription order includes one or more service components responsible for providing the services requested by a customer in the subscription order.

Additionally, TAS module 204 may also interact with one or more additional databases such as a Tenant Information System (TIS) database 320 to enable the provisioning of resources for one or more services subscribed by the customer while taking into consideration historical information, if any, available for the customer. TIS database 320 may include historical order information and historical usage information pertaining to orders subscribed by the customer.

TAS module 204 may be deployed using different deployment models. In certain embodiments, the deployment includes a central component that interfaces with one or more distributed components. The distributed components may, for example, be deployed as various data centers and accordingly may also refer to as data center components. The central component includes capabilities to process orders and coordinate services in cloud infrastructure system 100, while the data center components provide capabilities for provisioning and operating the runtime system that provides the resources for the subscribed services.

FIG. 4 depicts an exemplary distributed deployment of the TAS module, according to an embodiment of the present invention. In the embodiment depicted in FIG. 4, the distributed deployment of TAS module 204 includes a TAS central component 400 and one or more TAS Data Centers (DCs) components 402, 404 and 406. These components may be implemented in hardware, software, or combinations thereof.

In one embodiment, the responsibilities of TAS central component 400 include, without limitation, to provide a centralized component for receiving customer orders, performing order-related business operations such as creating a new subscription, changing the service level for a service in the subscription, changing the services included in the subscription, and extending the time period of the subscription, or canceling the subscription. The responsibilities of TAS central component 400 may also include maintaining and serving subscription data needed by cloud infrastructure system 100 and interfacing with order management module 214, support UI 216, cloud UI 212 and store UI 210 to handle all the back-office interactions.

In one embodiment, the responsibilities of TAS DCs 402, 404 and 406 include, without limitation, performing runtime operations for orchestrating the provisioning the resources for one or more services subscribed by the customer. TAS DCs 402, 404 and 406 also include capabilities to perform operations such as locking, unlocking, enabling, or disabling a subscription order, collecting metrics related to the order, determining the status of the order, and sending notification events related to the order.

In an exemplary operation of the distributed TAS system shown in FIG. 4, TAS central component 400 initially receives an order from a customer via cloud UI 212, store UI 210, via order management system 214, or via order database 224. In one embodiment, the customer represents a buyer who has financial information and the authority to order and/or change a subscription. In one embodiment, the order information includes information identifying the customer, the type of services that the customer wishes to subscribe to, and an account administrator who will be responsible for handling the request. In certain embodiments, the account administrator may be nominated by the customer when the customer places an order for a subscription to one or more services offered by cloud infrastructure system 100. Based on the order information, the TAS central component 400 identifies the data region of the world such as Americas, EMEA, or Asia Pacific in which the order originates and the particular TAS DCs (for e.g., 402, 404 or 406) that will be deployed for provisioning the order. In one embodiment, the particular TAS DC (for e.g., from among DCs 402, 404 or 406) that will be deployed for provisioning the order is determined based on the geographical data region in which the request originated.

TAS central component 400 then sends the order request to the particular TAS DC in which to provision services for the order request. In one embodiment, TAS DC’s 402, 404 or 406 identify a service administrator and an identity domain administrator responsible for processing the order request at the particular TAS DC. The service administrator and the identity administrator may be nominated by the account administrator identified in the subscription order. TAS DC’s 402, 404 or 406 communicate with SDI module 204 to orchestrate the provisioning of physical resources for the order. SDI component 204 in respective TAS DCs 402, 404 or 406, allocates resources and configures those resources needed to fulfill the subscription order.

In certain embodiments, TAS DCs, 402, 404 or 406 identify an identity domain associated with the subscription. SDI component 206 may provide the identity domain information to IDM component 200 (shown in FIG. 2) for identifying an existing identity domain or creating a new identity domain. Once the order is provisioned by the SDI module at respective TAS DCs, 402, 404 or 406, TAS central component 400 may place information regarding the provisioned resources in a support system, via support UI 216. Information may include, for example, displaying resource metrics related to the services and usage statistics of the services.

Once in operation, at each data center, EM module 208 periodically collects usage statistics for each of the provisioned services provisioned at that data center, such as the amount of storage used, the amount data transferred, the number of users, and the amount of system up time and system down time. These statistics are provided to the TAS DC that is local to EM module 208 (i.e., at the same data center). In an embodiment, the TAS DCs may use the usage statistics to determine whether overuse of a service has occurred, and if so, to determine how much to bill for the overage, and provide the billing information to order management system 214.

FIG. 5 is a simplified block diagram illustrating the interactions of the SDI module with one or more modules in the cloud infrastructure system, in accordance with an embodiment of the present invention. In one embodiment, SDI module 206 interacts with TAS module 204 to provision resources for services in a subscription order received by TAS
module 204. In certain embodiments, one or more of the modules illustrated in FIG. 5 may be modules within cloud infrastructure system 100. In other embodiments, one or more of the modules that interact with SDI module 206 may be outside cloud infrastructure system 100. In addition, alternative embodiments may have more or less modules than those shown in FIG. 5. These modules may be implemented in hardware, or software, or combinations thereof.

[0094] In one embodiment, the modules in SDI module 206 may include one or more modules in SaaS platform 102 and PaaS platform 104 in cloud infrastructure system 100. In order to perform provisioning of resources for various services, SDI module 206 may interact with various other modules, each customized to help with provisioning resources for a particular type of service. For example, as illustrated in FIG. 5, SDI module 206 may interact with a Java service provisioning control module 500 to provide Java cloud services. In one embodiment, Java service provisioning control component 500 may deploy a Java Cloud Service (JCS) assembly specified by SDI module 206 that includes a set of tasks to be performed to provision Java cloud services. Infrastructure resources 106 then determines the resources needed to provision the Java cloud services.

[0095] As other examples, SDI module 206 may interact with one or more modules such as a Virtual Assembly Builder (VAB) module 502, an Application Express (APEX) deployer module 504, a Virtual Machine (VM) module 506, an IDM module 208, and a database machine module 118. VAB module 502 includes capabilities to configure and provision complete multi-tier application environments. In one embodiment, VAB module 502 deploys a Middleware (MW) service assembly specified by SDI module 206 to provide an MW service in cloud infrastructure system 100 using the services provided by VM module 506. APEX deployer module 504 includes capabilities to configure and provision database services. In one embodiment, APEX deployer module 504 deploys a database service assembly specified by SDI module 206 to provide a database service in cloud infrastructure system 100 using the resources provided by infrastructure resources 106. SDI module 206 interacts with IDM module 208 to provide identity services such as access management across multiple applications in cloud infrastructure system 100.

[0096] FIG. 6 depicts a simplified high level diagram of sub-modules of the SDI module according to an embodiment of the present invention. In the embodiment depicted in FIG. 6, SDI module 206 includes a SDI-Web Services (WS) module 600, an SDI request controller module 602, an SDI task manager module 604, an SDI monitoring module 606, an SDI data access module 608, an SDI common library module 610, and an SDI connector module 612. These modules may be implemented in hardware, or software, or combinations thereof. SDI module 206 depicted in FIG. 6 and its various modules are meant for illustrative purposes only and are not intended to limit the scope of embodiments of the present invention. Alternative embodiments may have more or less modules than those shown in FIG. 6. These modules and their functions are described in detail below.

[0097] SDI-WS module 600 includes capabilities for receiving a step in the business process associated with an order from a web service processing layer (for example, via System Provisioning API discussed in FIG. 3B) in the form of a SOAP request to SDI-WS module 600. SDI request controller module 602 is the internal request processing engine in SDI module 206 and includes capabilities for performing asynchronous request processing, concurrent request processing, concurrent task processing, fault tolerant and recovery and plug-in support related to the order requests. In one embodiment, SDI request controller module 602 accepts each step of the business process associated with the order from SDI-WS module 600 and submits the step to SDI task manager module 604.

[0098] SDI task manager module 604 translates each step specified in the business process into a series of tasks for provisioning the particular step. Once the set of tasks for a specific step have been provisioned, SDI task manager module 604 responds to a business process executor 316 in TAS module 204 with operation results that includes an order payload with details of the resources provisioned to fulfill the particular step. SDI task manager module 604 repeats this process until all the steps of the particular business process associated with the order are complete.

[0100] In certain embodiments, SDI task manager module 604 translates each step specified in the business process into a series of tasks by utilizing the services of SDI connector module 612. SDI connector module 612 includes one or more connectors for handling the deployment of tasks specified by SDI task manager module 604 to provision one or more services related to the order request. In certain embodiments, one or more of the connectors may handle tasks that are specific to a particular service type while other connectors may handle tasks that are common across different service types. In one embodiment, SDI connector module 612 includes a set of connectors (wrapper APIs) that interface with one or more of the external modules (shown in FIG. 5) in cloud infrastructure system 100 to provision the services and resources related to the order request. For example, Application Express (APEX) connector 614 interfaces with APEX deployer module 504 to provision database services, Web Center Connector 616 (WCC) interfaces with a web center module in cloud infrastructure system 100 to provision web services. The web center module is a user engagement platform and includes capabilities for delivering connectivity between people and information in cloud infrastructure system 100.

[0101] In certain embodiments, Middleware Applications (MA) connector 618 interfaces with VAB module 502 in cloud infrastructure system 100 to provision middleware application services. NUVAIQ connector 620 interfaces with VAB module 502 to provision Java services. IDM connector 622 interfaces with IDM module 208 to provision identity and access management for users subscribing to services and resources in cloud infrastructure system 100. Virtual Assembly Builder (VAB) connector 624 interfaces with VAB module 502 in cloud infrastructure system 100 to configure and provision complete multi-tier application environments. Plug-in connector 626 interfaces with EM module 208 to manage and monitor the components in cloud infrastructure system 100. HTTP server connector 628 interfaces with one or more web servers in the PaaS platform to provide connectivity to users in cloud infrastructure system 100.

[0102] SDI monitoring module 606 in SDI module 206 provides an inbound interface for receiving Java Management
Extensions (JMX) requests. SDI monitoring module 606 also provides tools for managing and monitoring applications, system objects and devices in cloud infrastructure system 100. SDI-data access module 608 provides an inbound interface for receiving Java Database Connectivity (JDBC) requests. SDI-data access module 608 supports data access and provides object relational mapping, java transaction API services, data access objects, and connection pooling in cloud infrastructure system 100. The SDI-common library module 610 provides configuration support for the modules in SDI module 206.

[0103] The embodiment of FIG. 6 discussed above describes modules in the SDI module according to an embodiment of the present invention. FIG. 7A depicts a simplified flowchart 700 depicting processing that may be performed by the modules of the SDI module in the cloud infrastructure system, in accordance with an embodiment of the present invention. The processing depicted in FIG. 7A may be implemented in software (e.g., code, instructions, program) executed by one or more processors, hardware, or combinations thereof. The software may be stored in memory (e.g., on a memory device, on a non-transitory computer-readable storage medium). The particular series of processing steps depicted in FIG. 7A is not intended to be limiting. Other sequences of steps may also be performed according to alternative embodiments. For example, alternative embodiments of the present invention may perform the steps outlined above in a different order. Moreover, the individual steps illustrated in FIG. 7A may include multiple sub-steps that may be performed in various sequences as appropriate to the individual step. Furthermore, additional steps may be added or removed depending on the particular applications. One of ordinary skill in the art would recognize many variations, modifications, and alternatives. In one embodiment, the processing depicted in FIG. 7A may be performed by one or more modules in the SDI module 206 discussed in detail in FIG. 6.

[0104] At 702, a business process associated with a subscription order is received. In one embodiment, SDI-WS module 600 in SDI module 206 receives one or more steps in the business process associated with the subscription order from business process executor 316. At 704, each step in the business process is translated into a series of tasks for provisioning resources for the subscription order. In one embodiment, SDI task manager module 604 in SDI module 206 translates each step specified in the business process into a series of tasks by utilizing the services of SDI connector module 612. At 706, the subscription order is provisioned based on the series of tasks. In one embodiment, and as discussed in FIG. 6, SDI connector module 612 includes one or more connectors for handling the deployment of tasks specified by SDI task manager module 604 to provision resources for the services in the subscription order.

[0105] As described above with respect to FIG. 6, SDI task manager module 604 translates each step specified in a business process into a series of tasks by utilizing the services of SDI connector module 612, which may include one or more connectors for handling the deployment of tasks specified by SDI task manager module 604 to provision one or more services related to the order request. One or more of the connectors may handle tasks that are specific to a particular service type while other connectors may handle tasks that are common across different service types. In one embodiment, SDI connector module 612 includes a set of connectors (wrapper APIs) that interface with one or more of the external modules (shown in FIG. 5) in cloud infrastructure system 100 to provision the services and resources related to the order request. For example, a NUVAIQ connector 620 interfaces with VAB module 502 to provision Java services.

[0106] FIG. 7B depicts a simplified block diagram showing the high-level architecture of a Nuviaq system 710 and its relationships with other cloud infrastructure components according to an embodiment of the present invention. It should be appreciated that Nuviaq system 710 depicted in FIG. 7B may have other components than those depicted in FIG. 7B. Further, the embodiment shown in FIG. 7B is only one example of a cloud infrastructure system that may incorporate an embodiment of the invention. In some other embodiments, Nuviaq system 710 may have more or fewer components than shown in FIG. 7B, may combine two or more components, or may have a different configuration or arrangement of components.

[0107] In certain embodiments, Nuviaq system 710 may be configured to provide a runtime engine for orchestrating PaaS operations. Nuviaq system 710 may provide a web service API to facilitate integration with other products and services. Nuviaq system 710 also provides support for complex workflows in system provisioning, application deployment and associated lifecycle operations and integrates with management and monitoring solutions.

[0108] In the embodiment depicted in FIG. 7B, Nuviaq system 710 comprises a Nuviaq proxy 712, a Nuviaq manager 714, and a Nuviaq database 716. In certain embodiments, Nuviaq manager 714 provides an entry point into Nuviaq system 710, providing secure access to PaaS operations via the web service API. Internally, it tracks system state in the database and controls job execution on the workflow engine. In a public cloud, Nuviaq manager 714 may be accessed by the Tenant Provisioning system (SDI 206) and the Tenant Console, to drive provisioning and deployment operations respectively.

[0109] In one embodiment, Nuviaq manager 714 executes jobs asynchronously via an internal workflow engine. A job may be a sequence of actions specific to a given PaaS workflow. Actions may be performed in order, with failure in any step resulting in failure of the overall job. Many workflow actions delegate to external systems relevant to the workflow, such as the EM command line interface (CLI). In one implementation, Nuviaq manager 714 application may be hosted in a 2-node WebLogic cluster with associated HTTP server (e.g., Oracle HTTP Server or OHS) instance, running inside a firewall.

[0110] In certain embodiments, Nuviaq proxy 712 is the public access point to the Nuviaq API. In one embodiment, only Public API may be exposed here. Requests received by proxy 712 may be forwarded to Nuviaq manager 714. In one embodiment, Nuviaq proxy 712 runs outside the firewall, whereas manager 714 runs within the firewall. In one implementation, Nuviaq proxy 712 application runs on a WebLogic cluster running outside the firewall.

[0111] In certain embodiments, Nuviaq database 716 tracks various domain entities such as, without limitation, platform instance, deployment plan, application, WebLogic domain, jobs, alerts, and so on. Primary keys may be aligned with the Service Database where appropriate.

[0112] In one embodiment, Platform Instance 718 may contain all resources required for a WebLogic service for a given tenant.
Nuviaq system 710 may rely on additional systems of cloud infrastructure system 100 to carry out the workflows used the WebLogic cloud service. These dependencies may include dependencies on SDI 206, IDM 200, a virus scan system, a service database, CRM instances, and the like. For example, Nuviaq system 710 may depend upon functions performed by an Assembly Deployer in SDI 206. In one embodiment, the Assembly Deployer is a system to manage interactions with OVAAB (Oracle Virtual Assembly Builder) and OVM (Oracle Virtual Machine). Capabilities of the Assembly Deployer used by Nuviaq system 710 may include, without limitation, functions for deploying an assembly, undoing a deployed assembly, signing assembly deployment, scaling appliance, and the like. In one implementation, Nuviaq system 710 accesses the Assembly Deployer via a web service API.

In certain embodiments, security policies may require certain artifacts to be scanned for viruses before being deployed to an application. Cloud infrastructure system 100 may provide a virus scan system for this purpose that provides scanning as a service for multiple components of the public cloud.

In certain embodiments, a public cloud infrastructure may maintain a Service Database containing information about tenants (e.g., customers) and their service subscriptions. Nuviaq workflows may access this data in order to properly configure a WebLogic service as a client to other services that the tenant also subscribes to.

Nuviaq system 710 may depend on IDM 200 for its security integration. In certain embodiments, Java Service instances can be associated with a CRM instance. The association allows user applications deployed to their Java Service instance to access a CRM instance though Web Service calls.

Various entities may use services provided by Nuviaq system 710. These clients of Nuviaq system 710 may include: a Tenant Console, which is an management server (e.g., Oracle Management Server) based user interface that customers may access to manage their applications on their platform instances; several IDEs such as Oracle IDEs (JDeveloper, NetBeans, and OEPE) have been offered to offer access to application lifecycle management operations; one or more Command Line Interfaces (CLIs) that are available to access the platform instance the platform instance.

Provisioning use case for Nuviaq system 710 -- A Provision Platform Instance use case is realized via the Create Platform Instance operation of the Nuviaq API. In the context of cloud infrastructure system 100, a service instance with respect to the Nuviaq system corresponds to a Nuviaq platform instance. A platform instance is assigned a unique identifier is used on all subsequent operations related to this instance. A Platform Deployement descriptor provided to the Create Platform Instance action allows for properties to be set that modify the configuration of the platform instance to meet the subscription requirements of the tenant. These properties may include for example:

Property #1: oracle.cloud.service.weblogic.size
Values: BASIC, STANDARD, ENTERPRISE
Description: Specifies the subscription type. This impacts the number of servers, database limits and quality of service settings.

Property #2: oracle.cloud.service.weblogic.trial
Values: TRUE, FALSE
Description: Indicates whether or not this is a trial subscription.

Property #3: oracle.cloud.service.weblogic.crm
Values: CRM Service ID
Description: Identifies a CRM service to be associated with this WebLogic service instance.

FIG. 7C depicts an example sequence diagram illustrating steps of a provisioning process using a Nuviaq system according to an embodiment of the present invention. The sequence diagram depicted in FIG. 7C is only an example and is not intended to be limiting.

Install/Update Application use case — The Install Application operation deploys an application to a running WebLogic Server after validating that the application archive meets the security requirements of the Public Cloud. In one embodiment, the Application Deployment descriptor provided to the Install Application action for properties to be set that modify the configuration of the application to meet the subscription requirements of the tenant. These properties may include for example:

Property: oracle.cloud.service.weblogic.state
Values: RUNNING, STOPPED

Description: Specifies the initial state of the application after deployment.

FIG. 7D depicts an example sequence diagram illustrating steps of a deployment process using a Nuviaq system according to an embodiment of the present invention. The sequence diagram depicted in FIG. 7D is only an example and is not intended to be limiting.

Referring back to FIG. 2, in certain embodiments, TAS 204 and SDI 206 working in cooperation are responsible for provisioning resources for one or more services ordered by a customer from a set of services offered by cloud infrastructure system 100. For example, in one embodiment, for provisioning a database service, the automated provisioning flow may be as follows for a paid subscription:

1. Customer places an order for a paid subscription to a service via UI 210.
2. TAS 204 receives the subscription order.
3. When services are available TAS 204 initiates provisioning by using the services of SDI 206. TAS 204 may perform business process orchestration, which will execute the relevant business process to complete the provisioning aspect of the order. In one embodiment, TAS 204 may use a BPEL (Business Process Execution Language) Process Manager to orchestrate the steps involved in the provisioning and handle the lifecycle operations.
4. In one embodiment, to provision a database service, SDI 206 may call PL/SQL APIs in the CLOUD_UI to associate a schema for the requesting customer.
5. After successful association of a schema to the customer, SDI signals TAS and TAS send a notification to the customer that the database service is now available for use by the customer.
6. The customer may log into cloud infrastructure system 100 (e.g., using an URI such as cloud.oracle.com) and activate the service.

In some embodiments, a customer may also be allowed to subscribe to a service on a trial basis. For example, such a trial order may be received via cloud UI 212 (e.g., using cloud.oracle.com).

In certain embodiments, cloud infrastructure system 100 enables underlying hardware and service instances to be shared between customers or tenants. For example, the database service may be provisioned as shown in FIG. 7E in one
embodiment. FIG. 7E depicts multiple Exadata compute nodes 730 and 732, each providing a database instance provisioned for the database service. For example, compute node 730 provides a database instance 734 for a database service. Each Exadata compute node may have multiple database instances.

In certain embodiments, each database instance can comprise multiple schemas and the schemas may be associated with different customers or tenants. For example, in FIG. 7E, database instance 734 provides two schemas 736 and 738, each with its own tables. Schema 736 may be associated with a first customer or tenant subscribing to a database service and schema 738 may be associated with a second customer or tenant subscribing to the database service. Each tenant gets a completely isolated schema. Each schema acts like a container that can manage database objects including tables, views, stored procedures, triggers, etc., for the associated tenant. Each schema may have one dedicated tablespace, with each tablespace having one data file.

In this manner, a single database instance can provide database services to multiple tenants. This not only enables sharing of underlying hardware resources but also enables sharing of service instance between tenants.

In certain embodiments, such a multi-tenancy system is facilitated by IDM 200, which beneficially enables multiple separate customers, each having their own separate identity domains, to use hardware and software that is shared in the cloud. Consequently, there is no need for each customer to have its own dedicated hardware or software resources, and in some cases resources that are not being used by some customers at a particular moment can be used by other customers, thereby preventing those resources from being wasted. For example, as depicted in FIG. 7E, a database instance can service multiple customers each with their respective identity domains. Although each such database service instance can be a separate abstraction or view of a single physical multi-tenant database system that is shared among the many separate identity domains, each such database service instance can have a separate and potentially different schema than each other database service instance has. Thus, the multi-tenant database system can store mappings between customer-specified database schemas and the identity domains to which those database schemas pertain. The multi-tenant database system can cause the database service instance for a particular identity domain to use the schema that is mapped to that particular identity domain.

The multi-tenancy can also be extended to other services such as the Java Service. For example, multiple customers can have a JAVA service instance placed within their respective identity domains. Each such identity domain can have a JAVA virtual machine, which can be viewed as being a virtual “slice” of hardware. In one embodiment, a job-monitoring service (e.g., Hudson) can be combined with a JAVA enterprise edition platform (e.g., Oracle WebLogic) in the cloud to enable each separate identity domain to have its own separate virtual “slice” of the JAVA enterprise edition platform. Such a job-monitoring service can, for example, monitor the execution of repeated jobs, such as building a software project or jobs run by an operating system’s time-based job scheduler. Such repeated jobs can include the continuous building and/or testing of software projects. Additionally or alternatively, such repeated jobs can include the monitoring of executions of operating system-run jobs that are executed on machines that are remote from the machine on which the job-monitoring service executes.

In certain embodiments, techniques for automating the provisioning, managing and tracking of services provided by cloud infrastructure system 100 include providing the customer with the ability to request for services in a subscription order with different service levels for different time periods. As an example, a customer may request for a first level of a service for a first time period (e.g., for the first year) and a second level of the service for a second time period following the first time period (e.g., for the second year) and a third level of service for a third period of time following the second period of time, such as a third year and so on. The second level of service and the third level of service may be less than or more than the first level of service. In this manner, a customer is provided with the ability to subscribe to a multi-year (or other time period) plan, with changing service levels (increasing/decreasing) from one year to the next. As another example, the customer may also request for a first level of a service for a first time period and request for additional services at the same service level or at different service levels for future time periods.

The manner in which the service levels are defined may be service-specific and may vary from one service to another. In one embodiment, the service levels relate to the amount of resources assigned to one or more services in the subscription order. For example, a basic service level may provide a minimum level of storage, data transmission, or number of users, and higher service levels may include additional resources. As an example, for a CRM service, the level of the service may depend upon the number of users for which the service provides. A subscription order from a customer may request a CRM service for 100 users in the first year with an increase of another 100 users for the next year (i.e., for a total of 200 users in the second year), and another 200 users for the next year (i.e., for a total of 400 users in the third year), and so forth. It is to be appreciated that the service levels and time periods listed above are merely provided by way of example and are not intended to limit the scope of the present invention. One of ordinary skill in the art would recognize many variations, modifications, and alternatives. Certain embodiments of the present invention provide techniques for performing a phased deployment of a customer’s subscription order based on the service levels and time periods specified by the customer in the subscription order, in an efficient manner as will be discussed in detail below.

FIG. 8 is a high level block diagram illustrating modules in the cloud infrastructure system for performing a phased deployment of a customer’s subscription order, in accordance with one embodiment of the present invention. In one embodiment, processing related to performing a phased deployment of the customer’s subscription order is performed by one or more modules in order processing system 310 of TAS component 204, shown in FIG. 8. These modules include a provisioning order generator 800, an order scheduler 802 and a heuristics module 804. These modules may be implemented in hardware, or software, or combinations thereof. The modules depicted in FIG. 8 are meant for illustrative purposes only and are not intended to limit the scope of embodiments of the present invention. For example, alternative embodiments of the present invention may include additional modules or even modules that are not part of order processing system 310.
Upon receiving a subscription order from one or more input sources as discussed in FIG. 3B, in one embodiment, provisioning order generator 800 in the order processing system 310 analyzes the received subscription order to determine whether the subscription order includes a request for one or more services at different service levels for different time periods and generates separate orders for each of the identified time periods.

In one embodiment, the separate orders generated by provisioning order generator 800 may include a base order (also referred to as an initial order or an onboarding order) for a base time period, and one or more additional orders for future time periods. The base time period is the earliest time period for when the service is to be provided and the future time periods correspond to time periods after the base time period. In one embodiment, the additional orders may include one or more ramp-up orders or one or more ramp-down orders. Whether an order is a ramp-up order or a ramp-down order depends upon the level of service specified in the time period immediately prior to the ramp-up or ramp-down order. For example, if a subscription order requests a first level of service for a first time period and a second level of service for a second time period following the first time period, then if the second service level is higher than the first level, then the order for the second time period is a ramp-up order since the service level has been increased from the previous time period. However, the second level of service is lower or less than the first level of service, then the order generated for the second time period is considered a ramp-down order since the service level has been decreased from the immediately prior time period. In one embodiment, the separate orders generated by provisioning order generator 800 may also include generating multiple base orders for different services in the subscription order.

In certain embodiments, provisioning order generator 800 initially analyzes the subscription order to identify a request for a service at a first service level for a first time period. Provisioning order generator 800 then determines a base order for the service at the first service level for the first time period. Provisioning order generator 800 then inputs the base order into order scheduler 802. In one embodiment, order scheduler 802 schedules the base order for execution at the first time period.

In certain embodiments, provisioning order generator 800 also analyzes the subscription order to identify requests for additional service levels for the service for future time periods. Provisioning order generator 800 may also analyze the subscription order to identify requests for additional services with the same service level or with a different service level as compared to the first service level for future time periods. Provisioning order generator 800 then determines one or more additional orders for the additional service levels or the additional services for future time periods. The additional orders may include one or more ramp-up orders or one or more ramp-down orders. As an example, consider that provisioning order generator 800 identifies a request for a second service level for the service for a future time period. In such a case, in addition to the base order, provisioning order generator 800 generates a ramp-up order or a ramp-down order for the service based on the second service level for the future time period. As another example, consider that provisioning order generator 800 identifies a request for an additional service at the same service level or at a different service level for a future time period. In such a case, provisioning order generator 800 generates either a ramp-up order or a ramp-down order for the additional service for the future time period, in addition to the base order.

In one embodiment, as discussed above, provisioning order generator 800 may determine the additional orders based on identifying requests for additional service levels for different time periods or additional services for future time periods in the subscription order. In certain situations, a customer’s subscription order request may not include additional service levels or additional services for future time periods. In such a case, in an alternate embodiment, heuristics module 804 may automatically determine one or more additional orders for the customer based on heuristic information related to the customer. In one example, the heuristic information relates to the customer level of the customer specified in the subscription order. For example, if the customer level specified in the subscription order indicates that the customer is a gold customer based on the quality of service that the cloud infrastructure system guarantees to the customer (e.g., determined based on the customer’s SLA with cloud infrastructure system 100), heuristics module 804 may automatically determine an additional order which may be a ramp-up order for the customer for a future time period.

In another example, the heuristic information may also relate to the current usage of resources utilized by the customer. Accordingly, in another embodiment, heuristics module 804 may determine one or more ramp-up orders or ramp-down orders based on the current usage of computing resources in the base order. For example, if the current usage of resources in the base order indicates that a significant ramp-up of resources is expected in a future time period, then heuristics module 804 determines a ramp-up order with additional computing resources such as by assigning a high capacity server pool to the ramp-up order. On the other hand, if the current usage of resources in the base order indicates that a significant ramp-down of resources is expected in a future time period, then heuristics module 804 may determine a ramp-down order with fewer computing resources such as by assigning low capacity server pool to the ramp-down order. In one embodiment, the current usage of resources for the customer may be determined by accessing usage information stored in the TIS database 320.

In certain embodiments, the initial physical configuration of the subscription of the base order may be based on the ramp-up orders and the ramp-down orders specified by the customer. In one embodiment, the provisioning order generator may optimize the initial physical layout of the subscription based on the ramp-up order and the ramp-down orders specified by the customer as follows. For example, provisioning order generator 800 may assign the base subscription order initially to a high-capacity server pool if a significant ramp-up order in the near future is in the initial subscription order submitted by the customer. Later, provisioning order generator 800 may move the base subscription order to a low-capacity server pool if a significant ramp-down of resources is expected at a future time.

The order scheduler 802 then schedules the additional orders which may include the ramp-up orders and ramp-down orders for execution based on the future time periods. In one embodiment, order scheduler 802 outputs a temporally ordered queue in which each ramp-up order and ramp-down order is scheduled for execution in accordance with an execution date specified in the ramp-up and ramp-down orders. In the example of the CRM service discussed
above, a base order for a CRM service for 100 users may be executed for the first year, while a ramp-up order for the CRM service for 200 users may be queued and scheduled to be executed after the first year. Similarly, a future ramp-up order for providing CRM services for 400 users may be queued and scheduled to be executed two years from the original order date.

A business process associated with the base order, the ramp-up orders and the ramp-down orders are then identified by the business process identifier 312. The business process executor 316 executes the business process associated with the orders. As discussed in FIG. 3B, the business process executor 316 accomplishes this by sending a request to the SDI component 206 to enable the provisioning of resources needed to fulfill the base order, the ramp-up orders and the ramp-down orders.

In addition to specifying different service levels for different time periods, a customer may also specify updates to the subscription order including the base subscription order and the ramp-up or ramp-down orders at any time during the subscription. In this situation, the order scheduler 802 makes the appropriate updates to the queued entries including adding new entries or modifying or deleting existing entries.

The apparatus of FIG. 8 described above discusses an implementation of the order processing module in the cloud infrastructure system for performing a phased deployment of a customer’s subscription order. FIG. 9 depicts a simplified flowchart depicting processing that may be performed by the order processing module in the cloud infrastructure system, in accordance with an embodiment of the present invention. The processing 900 depicted in FIG. 9 may be implemented in software (e.g., code, instructions, program) executed by one or more processors, hardware, or combinations thereof. The software may be stored in memory (e.g., on a memory device, on a non-transitory computer-readable storage medium). The particular series of processing steps depicted in FIG. 9 is not intended to be limiting. Other sequences of steps may also be performed according to alternative embodiments. For example, alternative embodiments of the present invention may perform the steps outlined above in a different order. Moreover, the individual steps illustrated in FIG. 9 may include multiple sub-steps that may be performed in various sequences as appropriate to the individual step. Furthermore, additional steps may be added or removed depending on the particular applications. One of ordinary skill in the art would recognize many variations, modifications, and alternatives. In one embodiment, the processing depicted in FIG. 9 may be performed by one or more modules in the order processing system 310 discussed in detail in FIG. 8.

At 902, a subscription order requesting one or more services offered by cloud infrastructure system 100, is received. At step 904, the subscription order is analyzed to identify a request for a service at a first service level for a first time period.

At 906, a base order is scheduled for execution at the first time period. The base order corresponds to the service requested at the first service level for the first time period.

At 908, the subscription order is analyzed to identify requests for additional service levels for the service for future time periods. The subscription order may also be analyzed to identify requests for additional services with the same service level or with a different service level as compared to the first service level for future time periods.

At 910, one or more additional orders are scheduled for execution at the future time periods. In one embodiment, the additional orders correspond to requests for additional service levels to be provided of the service for the future time periods. In another embodiment, the additional orders may correspond to requests for additional services to be provided with the same service level or with a different service level as compared to the first service level for the future time periods. In one embodiment, the additional orders may include one or more ramp-up orders or one or more ramp-down orders. In another embodiment, as discussed in FIG. 8, the additional orders may be determined based on heuristic information related to the customer. In one example, the heuristic information relates to the customer level of the customer specified in the subscription order. In another example, the heuristic information may also relate to the current usage of resources utilized by the customer in the base order. In other embodiments, the additional orders may also specify removing one or more services in the future time periods.

FIG. 10 is a simplified block diagram of a computing system 1000 that may be used in accordance with embodiments of the present invention. For example, a computer-readable storage medium 100 may comprise one or more computing devices. System 1000 depicted in FIG. 10 may be an example of one such computing device. Computer system 1000 is shown comprising hardware elements that may be electrically coupled via a bus 1024. The hardware elements may include one or more central processing units (CPUs) 1002, one or more input devices 1004 (e.g., a mouse, a keyboard, etc.), and one or more output devices 1006 (e.g., a display device, a printer, etc.). The CPUs may include single or multicore CPUs. Computer system 1000 may also include one or more storage devices 1008. By way of example, the storage device(s) 1008 may include devices such as disk drives, optical storage devices, and solid-state storage devices such as a random access memory (RAM) and/or a read-only memory (ROM), which can be programmable, flash-updateable and/or the like.

Computer system 1000 may additionally include a computer-readable storage media reader 1012, a communications subsystem 1014 (e.g., a modem, a network card (wireless or wired), an infra-red communication device, etc.), and working memory 1018, which may include RAM and ROM devices as described above. In some embodiments, computer system 1900 may also include a processing acceleration unit 1016, which can include a digital signal processor (DSP), a special-purpose processor, and/or the like.

Computer-readable storage media reader 1012 can further be connected to a computer-readable storage medium 1010, together (and, optionally, in combination with storage device(s) 1008) comprehensively representing remote, local, fixed, and/or removable storage devices plus storage media for temporarily and/or more permanently containing computer-readable information. Communications system 1014 may permit data to be exchanged with network 1024 and/or any other computer described above with respect to system environment 1000.

Computer system 1000 may also comprise software elements, shown as being currently located within working memory 1018, including an operating system 1020 and/or other code 1022, such as an application program (which may be a client application, a Web browser, a mid-tier application, a RDBMS, etc.). In an exemplary embodiment, working
memory 1018 may include executable code and associated data structures such as memory structures used for processing authorization requests described above. It should be appreciated that alternative embodiments of computer system 1000 may have numerous variations from that described above. For example, customized hardware might also be used and/or particular elements might be implemented in hardware, software (including portable software, such as applets), or both. Further, connection to other computing devices such as network input/output devices may be employed.

[0158] Storage media and computer-readable media for containing code, or portions of code, can include any appropriate media known or used in the art, including storage media and communication media, such as but not limited to volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Examples of storage and computer-readable media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other memory medium which can be used to store the desired information and which can be read by a computer. Storage media and computer readable media may include non-transitory memory devices.

[0159] Although specific embodiments of the invention have been described, various modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the invention. Embodiments of the present invention are not restricted to operation within certain specific data processing environments, but are free to operate within a plurality of data processing environments. Additionally, although embodiments of the present invention have been described using a particular series of transactions and steps, it should be apparent to those skilled in the art that the scope of the present invention is not limited to the described series of transactions and steps.

[0160] Further, while embodiments of the present invention have been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the present invention. Embodiments of the present invention may be implemented only in hardware, or only in software, or using combinations thereof.

[0161] The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that additions, subtractions, deletions, and other modifications and changes may be made thereunto without departing from the broader spirit and scope as set forth in the claims.

That which is claimed is:
1. A system comprising:
   one or more computing devices configurable to provide one or more services;
   a memory configurable to store subscription order information identifying a service from the set of services; and wherein a computing device from the one or more computing devices is configurable to:
   receive a subscription order requesting the service;
   identify, based upon the subscription order, a request for a first service level to be provided for the service for a first time period;
   cause a base order to be scheduled for execution for the first time period, the base order corresponding to the service at the first service level for the first time period;
   identify based upon the subscription order, a request for one or more additional service levels to be provided for the service for one or more future time periods, wherein the one or more future time periods is after the first time period; and
   cause one or more additional orders to be scheduled for execution for the one or more future time periods, the one or more additional orders corresponding to the one or more additional service levels of the service to be provided over the one or more future time periods.
2. The system of claim 1 wherein the computing device is configured to identify, based upon the subscription order, a request for one or more additional services to be provided with the same service level or with a different service level as compared to the first service level at the one or more future time periods.
3. The system of claim 1 wherein the one or more additional orders comprise one or more ramp-up orders, wherein the one or more ramp-up orders specify a service level that is higher than the first service level corresponding to the base order.
4. The system of claim 1 wherein the one or more additional orders comprise one or more ramp-down orders, wherein the one or more ramp-down orders specify a service level that is lower than the first service level corresponding to the base order.
5. The system of claim 1 wherein the one or more additional orders are determined based on a customer level related to the customer specified in the subscription order.
6. The system of claim 1 wherein the one or more additional orders are determined based on a current usage of computing resources utilized by the customer in the base order.
7. The system of claim 1 wherein the computing device is configured to determine an initial physical configuration of the base order based on the one or more additional orders.
8. A method comprising:
   receiving, by a computing device in a cloud infrastructure system, a subscription order requesting a service offered by the cloud infrastructure system;
   identifying, by the computing device, based upon the subscription order, a request for a first service level to be provided for the service for a first time period;
   causing, by the computing device, a base order to be scheduled for execution for the first time period, the base order corresponding to the service at the first service level for the first time period;
   identifying, by the computing device, based upon the subscription order, a request for one or more additional service levels to be provided for the service for one or more future time periods, wherein the one or more future time periods is after the first time period;
   causing, by the computing device, one or more additional orders to be scheduled for execution for the one or more future time periods, the one or more additional orders corresponding to the one or more additional service levels of the service to be provided over the one or more future time periods.
9. The method of claim 8 comprising identifying, based upon the subscription order, a request for one or more additional services to be provided with the same service level or
with a different service level as compared to the first service level at the one or more future time periods.

10. The method of claim 8 wherein the one or more additional orders comprise one or more ramp-up orders, wherein the one or more ramp-up orders specify a service level that is higher than the first service level corresponding to the base order.

11. The method of claim 8 wherein the one or more additional orders comprise one or more ramp-down orders, wherein the one or more ramp-down orders specify a service level that is lower than the first service level corresponding to the base order.

12. The method of claim 11 wherein the one or more ramp-down orders specify a removal of the service from the subscription order.

13. The method of claim 1 wherein the one or more additional orders are determined based on a customer level related to the customer specified in the subscription order.

14. The method of claim 1 wherein the one or more additional orders are determined based on a current usage of computing resources utilized by the customer in the base order.

15. A computer-readable memory storing a plurality of instructions executable by one or more processors, the plurality of instructions comprising:
   - instructions that cause at least one processor from the one or more processors to store subscription order information to identify a service from the set of services;
   - instructions that cause at least one processor from the one or more processors to receive a subscription order requesting the service;
   - instructions that cause at least one processor from the one or more processors to identify, based upon the subscription order, a request for a first service level to be provided for the service for a first time period;
   - instructions that cause at least one processor from the one or more processors to cause a base order to be scheduled for execution for the first time period, the base order corresponding to the service at the first service level for the first time period;
   - instructions that cause at least one processor from the one or more processors to identify based upon the subscription order, a request for one or more additional service levels to be provided for the service for one or more future time periods, wherein the one or more future time periods is after the first time period; and
   - instructions that cause at least one processor from the one or more processors to cause one or more additional orders to be scheduled for execution for the one or more future time periods, the one or more additional orders corresponding to the one or more additional service levels of the service to be provided over the one or more future time periods.

16. The computer-readable memory of claim 15 wherein the instructions that cause at least one processor from the one or more processors to identify the request comprise instructions to identify, based upon the subscription order, a request for one or more additional services to be provided with the same service level or with a different service level as compared to the first service level at the one or more future time periods.

17. The computer-readable memory of claim 15 wherein the one or more additional orders comprise one or more ramp-up orders, wherein the one or more ramp-up orders specify a service level that is higher than the first service level corresponding to the base order.

18. The computer-readable memory of claim 15 wherein the one or more additional orders comprise one or more ramp-down orders, wherein the one or more ramp-down orders specify a service level that is lower than the first service level corresponding to the base order.

19. The computer-readable memory of claim 15 wherein the one or more additional orders are determined based on at least a customer level related to the customer specified in the subscription order or based on a current usage of computing resources utilized by the customer in the base order.

20. The computer-readable memory of claim 15 further comprising instructions that cause at least one processor from the one or more processors to receive updates to one or more of the service levels and the time periods specified in the base order and the one or more additional orders from the customer and modifying the one or more service levels and the time periods based on the updates.

* * * * *