(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 August 2023 (17.08.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/154683 A1

(51) International Patent Classification:
G06F 9/50 (2006.01) *G06F 9/48* (2006.01)
G06F 8/60 (2018.01)

(21) International Application Number:
PCT/US2023/062060

(22) International Filing Date:
06 February 2023 (06.02.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

63/308,003	08 February 2022 (08.02.2022)	US
63/312,814	22 February 2022 (22.02.2022)	US
63/315,017	28 February 2022 (28.02.2022)	US
18/105,766	03 February 2023 (03.02.2023)	US

(71) Applicant: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, M/S 5OP7, Redwood Shores, California 94065 (US).

(72) Inventors: **MILLER, Erik Joseph**; c/o ORACLE INTERNATIONAL CORPORATION, 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US). **BELLEAU, Michel**; c/o ORACLE INTERNATIONAL CORPORATION, 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US).

(74) Agent: **PLOVANIC, Jacob T.** et al.; 1100 Peachtree Street, Suite 2800, Mailstop: IP Docketing - 22, Atlanta, Georgia 30309 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

(54) Title: TECHNIQUES FOR MIGRATING SERVICES FROM A VIRTUAL BOOTSTRAP ENVIRONMENT

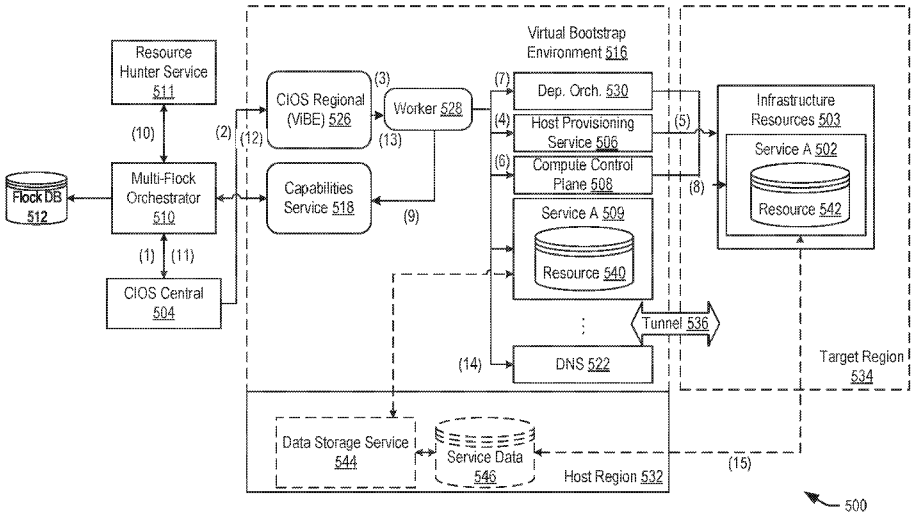


FIG. 5

(57) Abstract: Techniques are disclosed for migrating services from a virtual bootstrap environment. A distributed computing system can generate a virtual cloud network in a data center of a host region. A virtual bootstrap environment may be implemented in the virtual cloud network. The virtual bootstrap environment can include a plurality of services. The distributed computing system can also deploy an instance of one of the plurality of services to a target region data center. When the instance has been deployed, an indication that the deployment was successful can be received by the distributed computing system. In response, the distributed computing system may identify additional resources associated with the deployed instance of the service and update another service in the virtual bootstrap environment with that resource.

WO 2023/154683 A1

[Continued on next page]

HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

TECHNIQUES FOR MIGRATING SERVICES FROM A VIRTUAL BOOTSTRAP ENVIRONMENT

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Patent Application No. 18/105,766, filed February 3, 2023, entitled "TECHNIQUES FOR MIGRATING SERVICES FROM A VIRTUAL BOOTSTRAP ENVIRONMENT"; U.S. Provisional Patent Application No. 63/315,017, filed February 28, 2022, and entitled "TECHNIQUES FOR MIGRATING SERVICES FROM A VIRTUAL BOOTSTRAP ENVIRONMENT"; U.S. Provisional Patent Application No. 63/308,003, filed February 8, 2022, entitled "TECHNIQUES FOR BOOTSTRAPPING A REGION BUILD"; and U.S. Provisional Patent Application No. 63/312,814, filed February 22, 2022, entitled "TECHNIQUES FOR IMPLEMENTING VIRTUAL DATA CENTERS." The contents of each of these applications are hereby incorporated by reference in their entirety for all purposes.

TECHNICAL FIELD

[0002] This disclosure relates to building regional data centers. More specifically, this disclosure describes techniques for migrating services from a virtual bootstrap environment to data center infrastructure during the building of a regional data center.

BACKGROUND

[0003] Cloud infrastructure providers may offer cloud computing infrastructure and related services in numerous geographical areas worldwide. To provide this infrastructure, the cloud infrastructure provider may operate one or more data centers corresponding to a localized geographical area. These data centers may be included as part of a "region," a logical abstraction of the geographical area and the computing resources of the one or more data centers. Building new regions can include provisioning the computing resources, configuring infrastructure, and deploying code to those resources. Conventional techniques for building a region involve significant manual operations. Bootstrapping existing services into a new region may be challenging since a service may depend on the functionality of other existing services and/or resources in the region. Relying on manual operations to bootstrap services and/or build regions incurs substantial time costs, risks associated with manual configuration errors, and may not scale well.

BRIEF SUMMARY

[0004] Embodiments of the present disclosure relate to creating a bootstrapping environment to support building a region. The region build process can include bootstrapping (e.g., provisioning and/or deploying) resources (e.g., infrastructure components, artifacts, etc.) for any suitable number of services in a region (e.g., a geographical location associated with one or more data centers). The bootstrapping environment may be a virtual environment (e.g., a virtual cloud network) within an existing region. The virtual bootstrapping environment (ViBE) may therefore be constructed and configured in the existing region in advance of the region build process. Services (e.g., core services) may be deployed to the ViBE to support bootstrapping operations into a target region (e.g., the region to be built in the region build process). The services in the ViBE may be used to provision computing resources (e.g., bare metal computing hosts, virtual machines, storage, etc.) in the target region. The services in the ViBE may also be used to deploy services to the target region, including instances of the services in the ViBE. By using a cloud infrastructure orchestration service in conjunction with the ViBE, the new region may be built intelligently and automatically.

[0005] One embodiment is directed to a computer-implemented method performed by a distributed computing system (e.g., a cloud computing system) of a cloud service provider. The method may include generating a virtual cloud network in a data center of a host region and implementing a ViBE within the virtual cloud network. The ViBE can include a plurality of services. The method can also include deploying an instance of one of the plurality of the services in the ViBE to a target region data center. The instance may be configured to perform the same service functionality as the service in the ViBE. The method can also include receiving from the deployed instance an indication that the instance was successfully deployed. Since the deployed service may not have all of the resources associated with its counterpart service in the ViBE (e.g., data resources created by the service in the ViBE), in some embodiments, the indication may be a capability indicating a successful partial deployment of the service. The capability may be published to a capabilities service in the ViBE. The method can also include identifying a resource (e.g., a DNS record) associated with the instance deployed to the target region data center. The resource may then be used by the distributed computing system to update a second service in the ViBE (e.g., update a DNS in the ViBE). In some embodiments, the resource may also be used to update the instance of the service deployed to the target region data center.

[0006] Another embodiment is directed to a computing device comprising one or more processors and instructions that, when executed by the one or more processors, cause the computing device to perform the method(s) disclosed herein.

[0007] Still another embodiment is directed to a non-transitory computer-readable medium storing computer-executable instructions that, when executed by one or more processors of a computing cluster, cause the computing cluster to perform the method(s) disclosed herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

[0009] FIG. 1 is a block diagram of an environment in which a Cloud Infrastructure Orchestration Service (CIOS) may operate to dynamically provide bootstrap services in a region, according to at least one embodiment.

[0010] FIG. 2 is a block diagram for illustrating an environment and method for building a virtual bootstrap environment (ViBE), according to at least one embodiment.

[0011] FIG. 3 is a block diagram for illustrating an environment and method for bootstrapping services to a target region utilizing the ViBE, according to at least one embodiment.

[0012] FIG. 4 is a block diagram of an environment in which the Cloud Infrastructure Orchestration Service (CIOS) may utilize a Resource Hunter to discover resources during region build, according to at least one embodiment.

[0013] FIG. 5 is a block diagram depicting an example flow for executing operations for provisioning and deploying services to a data center of a new region and importing resources before updating a DNS record for the deployed service, according to at least one embodiment.

[0014] FIG. 6 illustrates an example method for migrating services from a ViBE to a target region, according to at least one embodiment.

[0015] FIG. 7 is a block diagram illustrating one pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

[0016] FIG. 8 is a block diagram illustrating another pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

[0017] FIG. 9 is a block diagram illustrating another pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

[0018] FIG. 10 is a block diagram illustrating another pattern for implementing a cloud infrastructure as a service system, according to at least one embodiment.

[0019] FIG. 11 is a block diagram illustrating an example computer system, according to at least one embodiment.

DETAILED DESCRIPTION

Example Data Center Build (Region Build) Infrastructure

[0020] The adoption of cloud services has seen a rapid uptick in recent times. Various types of cloud services are now provided by various different cloud service providers (CSPs). The term cloud service is generally used to refer to a service or functionality that is made available by a CSP to users or customers on demand (e.g., via a subscription model) using systems and infrastructure (cloud infrastructure) provided by the CSP. Typically, the servers and systems that make up the CSP's infrastructure, and which are used to provide a cloud service to a customer, are separate from the customer's own on-premises servers and systems. Customers can thus avail themselves of cloud services provided by the CSP without having to purchase separate hardware and software resources for the services. Cloud services are designed to provide a subscribing customer easy, scalable, and on-demand access to applications and computing resources without the customer having to invest in procuring the infrastructure that is used for providing the services or functions. Various different types or models of cloud services may be offered such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS), and others. A customer can subscribe to one or more cloud services provided by a CSP. The customer can be any entity such as an individual, an organization, an enterprise, and the like.

[0021] As indicated above, a CSP is responsible for providing the infrastructure and resources that are used for providing cloud services to subscribing customers. The resources provided by the CSP can include both hardware and software resources. These resources can include, for example, compute resources (e.g., virtual machines, containers, applications, processors), memory resources (e.g., databases, data stores), networking resources (e.g., routers, host machines, load balancers), identity, and other resources. In certain implementations, the resources provided by a CSP for providing a set of cloud services CSP are organized into data centers. A data center may be configured to provide a particular set of

cloud services. The CSP is responsible for equipping the data center with infrastructure and resources that are used to provide that particular set of cloud services. A CSP may build one or more data centers.

[0022] Data centers provided by a CSP may be hosted in different regions. A region is a localized geographic area and may be identified by a region name. Regions are generally independent of each other and can be separated by vast distances, such as across countries or even continents. Regions are grouped into realms. Examples of regions for a CSP may include US West, US East, Australia East, Australia Southeast, and the like.

[0023] A region can include one or more data centers, where the data centers are located within a certain geographic area corresponding to the region. As an example, the data centers in a region may be located in a city within that region. For example, for a particular CSP, data centers in the US West region may be located in San Jose, California; data centers in the US East region may be located in Ashburn, Virginia; data centers in the Australia East region may be located in Sydney, Australia; data centers in the Australia Southeast region may be located in Melbourne, Australia; and the like.

[0024] Data centers within a region may be organized into one or more availability domains, which are used for high availability and disaster recovery purposes. An availability domain can include one or more data centers within a region. Availability domains within a region are isolated from each other, fault tolerant, and are architected in such a way that data centers in multiple availability domains are very unlikely to fail simultaneously. For example, the availability domains within a region may be structured in a manner such that a failure at one availability domain within the region is unlikely to impact the availability of data centers in other availability domains within the same region.

[0025] When a customer or subscriber subscribes to or signs up for one or more services provided by a CSP, the CSP creates a tenancy for the customer. The tenancy is like an account that is created for the customer. In certain implementations, a tenancy for a customer exists in a single realm and can access all regions that belong to that realm. The customer's users can then access the services subscribed to by the customer under this tenancy.

[0026] As indicated above, a CSP builds or deploys data centers to provide cloud services to its customers. As a CSP's customer base grows, the CSP typically builds new data centers in new regions or increases the capacity of existing data centers to service the customers' growing demands and to better serve the customers. Preferably, a data center is built in close

geographical proximity to the location of customers serviced by that data center. Geographical proximity between a data center and customers serviced by that data center lends to more efficient use of resources and faster and more reliable services being provided to the customers. Accordingly, a CSP typically builds new data centers in new regions in geographical areas that are geographically proximal to the customers serviced by the data centers. For example, for a growing customer base in Germany, a CSP may build one or more data centers in a new region in Germany.

[0027] Building a data center (or multiple data centers) in a region is sometimes also referred to as building a region. The term "region build" is used to refer to building one or more data centers in a region. Building a data center in a region involves provisioning or creating a set of new resources that are needed or used for providing a set of services that the data center is configured to provide. The end result of the region build process is the creation of a data center in a region, where the data center is capable of providing a set of services intended for that data center and includes a set of resources that are used to provide the set of services.

[0028] Building a new data center in a region is a very complex activity requiring extensive coordination between various bootstrapping activities. At a high level, this involves the performance and coordination of various tasks such as: identifying the set of services to be provided by the data center; identifying various resources that are needed for providing the set of services; creating, provisioning, and deploying the identified resources; wiring the resources properly so that they can be used in an intended manner; and the like. Each of these tasks further have subtasks that need to be coordinated, further adding to the complexity. Due to this complexity, presently, the building of a data center in a region involves several manually initiated or manually controlled tasks that require careful manual coordination. As a result, the task of building a new region (i.e., building one or more data centers in a region) is very time consuming. It can take time, for example many months, to build a data center. Additionally, the process is very error prone, sometimes requiring several iterations before a desired configuration of the data center is achieved, which further adds to the time taken to build a data center. These limitations and problems severely limit a CSP's ability to grow computing resources in a timely manner responsive to increasing customer needs.

[0029] The present disclosure describes techniques for reducing build time, reducing computing resource waste, and reducing risk related to building one or more data centers in a

region. Instead of weeks and months needed to build a data center in a region in the past, the techniques described herein can be used to build a new data center in a region in a relatively much shorter time, while reducing the risk of errors over conventional approaches.

[0030] A Cloud Infrastructure Orchestration Service (CIOS) is disclosed herein that is configured to bootstrap (e.g., provision and deploy) services into a new data center based on predefined configuration files that identify the resources (e.g., infrastructure components and software to be deployed) for implementing a given change to the data center. The CIOS can parse and analyze configuration files (e.g., flock configs) to identify dependencies between resources, execution targets, phases, and flocks. The CIOS may generate specific data structures from the analysis and may use these data structures to drive operations and to manage an order by which services are bootstrapped to a region. The CIOS may utilize these data structures to identify when it can bootstrap a service, when bootstrapping is blocked, and/or when bootstrapping operations associated with a previously blocked service can resume. Advantageously, the CIOS can identify circular dependencies within the data structures and execute operations to eliminate/resolve these circular dependencies prior to task execution. Using these techniques, the CIOS substantially reduces the risk of executing tasks prior to the availability of the resources on which those tasks depend.

[0031] Utilizing the techniques disclosed herein, the CIOS may optimize parallel processing to execute changes to a data center while ensuring that tasks are not initiated until the functionality on which those tasks depend is available in the region. In this manner, the CIOS enables a region build to be performed more efficiently, which greatly reduces the time required to build a data center and the wasteful computing resource use found in conventional approaches.

Certain Definitions

[0032] A "region" is a logical abstraction corresponding to a geographical location. A region can include any suitable number of one or more execution targets. In some embodiments, an execution target could correspond to a data center.

[0033] An "execution target" refers to a smallest unit of change for executing a release. A "release" refers to a representation of an intent to orchestrate a specific change to a service (e.g., deploy version 8, "add an internal DNS record," etc.). For most services, an

execution target represents an "instance" of a service. A single service can be bootstrapped to each of one or more execution targets. An execution target may be associated with a set of devices (e.g., a data center).

[0034] "Bootstrapping" is intended to refer to the collective tasks associated with provisioning and deployment of any suitable number of resources (e.g., infrastructure components, artifacts, etc.) corresponding to a single service.

[0035] A "service" refers to functionality provided by a set of resources. A set of resources for a service includes any suitable combination of infrastructure, platform, or software (e.g., an application) hosted by a cloud provider that can be configured to provide the functionality of a service. A service can be made available to users through the Internet.

[0036] An "artifact" refers to code being deployed to an infrastructure component or a Kubernetes engine cluster, this may include software (e.g., an application), configuration information (e.g., a configuration file) for an infrastructure component, or the like.

[0037] A "flock config" refers to a configuration file (or a set of configuration files) that describes a set of all resources (e.g., infrastructure components and artifacts) associated with a single service. A flock config may include declarative statements that specify one or more aspects corresponding to a desired state of the resources of the service.

[0038] "Service state" refers to a point-in-time snapshot of every resource (e.g., infrastructure resources, artifacts, etc.) associated with the service. The service state indicates status corresponding to provisioning and/or deployment tasks associated with service resources.

[0039] IaaS provisioning (or "provisioning") refers to acquiring computers or virtual hosts for use, and even installing needed libraries or services on them. The phrase "provisioning a device" refers to evolving a device to a state in which it can be utilized by an end-user for their specific use. A device that has undergone the provisioning process may be referred to as a "provisioned device." Preparing the provisioned device (installing libraries and daemons) may be part of provisioning; this preparation is different from deploying new applications or new versions of an application onto the prepared device. In

most cases, deployment does not include provisioning, and the provisioning may need to be performed first. Once prepared, the device may be referred to as "an infrastructure component."

[0040] IaaS deployment (or "deployment") refers to the process of providing and/or installing a new application, or a new version of an application, onto a provisioned infrastructure component. Once the infrastructure component has been provisioned (e.g., acquired, assigned, prepared, etc.), additional software may be deployed (e.g., provided to and installed on the infrastructure component). The infrastructure component can be referred to as a "resource" after provisioning and deployment has concluded. Examples of resources may include, but are not limited to, virtual machines, databases, object storage, block storage, load balancers, and the like.

[0041] A "capability" identifies a unit of functionality associated with a service. The unit could be a portion, or all, of the functionality to be provided by the service. By way of example, a capability can be published indicating that a resource is available for authorization/authentication processing (e.g., a subset of the functionality to be provided by the resource). As another example, a capability can be published indicating the full functionality of the service is available. Capabilities can be used to identify functionality on which a resource or service depends and/or functionality of a resource or service that is available for use.

[0042] A "virtual bootstrap environment" (ViBE) refers to a virtual cloud network that is provisioned in the overlay of an existing region (e.g., a "host region"). Once provisioned, a ViBE is connected to a new region using a communication channel (e.g., an IPSec Tunnel VPN). Certain essential core services (or "seed" services) like a deployment orchestrator, a public key infrastructure (PKI) service, and the like can be provisioned in a ViBE. These services can provide the capabilities required to bring the hardware online, establish a chain of trust to the new region, and deploy the remaining services in the new region. Utilizing the virtual bootstrap environment can prevent circular dependencies between bootstrapping resources by utilizing resources of the host region. Services can be staged and tested in the ViBE prior to the physical region (e.g., the target region) being available.

[0043] A "Cloud Infrastructure Orchestration Service" (CIOS) may refer to a system configured to manage provisioning and deployment operations for any suitable number of services as part of a region build.

[0044] A Multi-Flock Orchestrator (MFO) may be a computing component (e.g., a service) that coordinates events between components of the CIOS to provision and deploy services to a target region (e.g., a new region). An MFO tracks relevant events for each service of the region build and takes actions in response to those events.

[0045] A "host region" refers to a region that hosts a virtual bootstrap environment (ViBE). A host region may be used to bootstrap a ViBE.

[0046] A "target region" refers to a region under build.

[0047] "Publishing a capability" refers to "publishing" as used in a "publisher-subscriber" computing design or otherwise providing an indication that a particular capability is available (or unavailable). The capabilities are "published" (e.g., collected by a capabilities service, provided to a capabilities service, pushed, pulled, etc.) to provide an indication that functionality of a resource/service is available. In some embodiments, capabilities may be published/transmitted via an event, a notification, a data transmission, a function call, an API call, or the like. An event (or other notification/data transmission/etc.) indicating availability of a particular capability can be broadcasted/addressed (e.g., published) to a capabilities service.

[0048] A "Capabilities Service" may be a flock configured to model dependencies between different flocks. A capabilities service may be provided within a Cloud Infrastructure Orchestration Service and may define what capabilities, services, and/or features have been made available in a region.

[0049] A "Real-time Regional Data Distributor" (RRDD) may be a service or system configured to manage region data. This region data can be injected into flock configs to dynamically create execution targets for new regions.

[0050] In some examples, techniques for implementing a Cloud Infrastructure Orchestration Service (CIOS) are described herein. Such techniques, as described briefly

above, can be configured to manage bootstrapping (e.g., provisioning and deploying software to) infrastructure components within a cloud environment (e.g., a region). In some instances, the CIOS can include computing components (e.g., a CIOS Central and a CIOS Regional, both of which will be described in further detail below) that may be configured to manage bootstrapping tasks (provisioning and deployment) for a given service and a Multi-Flock Orchestrator (also described in further detail below) configured to initiate/manage region builds (e.g., bootstrapping operations corresponding to multiple services).

[0051] The CIOS enables region building and world-wide infrastructure provisioning and code deployment with minimal manual run-time effort from service teams (e.g., beyond an initial approval and/or physical transportation of hardware, in some instances). The high-level responsibilities of the CIOS include, but are not limited to, coordinating region builds, providing users with a view of the current state of resources managed by the CIOS (e.g., of a region, across regions, world-wide, etc.), and managing bootstrapping operations for bootstrapping resources within a region.

[0052] The CIOS may provide view reconciliation, where a view of a desired state (e.g., a desired configuration) of resources may be reconciled with a current/actual state (e.g., a current configuration) of the resources. In some instances, view reconciliation may include obtaining state data to identify what resources are actually running and their current configuration and/or state. Reconciliation can be performed at a variety of granularities, such as at a service level.

[0053] The CIOS can perform plan generation, where differences between the desired and current state of the resources are identified. Part of plan generation can include identifying the operations that would need to be executed to bring the resources from the current state to the desired state. In some examples, the CIOS may present a generated plan to a user for approval. In these examples, the CIOS can mark the plan as approved or rejected based on user input from the user. Thus, users can spend less time reasoning about the plan and the plans are more accurate because they are machine generated. Plans are almost too detailed for human consumption; however, the CIOS can provide this data via a sophisticated user interface (UI).

[0054] In some examples, the CIOS can handle execution of change management by executing the approved plan. Once an execution plan has been created and approved, engineers may no longer need to participate in change management unless the CIOS initiates roll-back. The CIOS can handle rolling back to a previous service version by generating a plan that returns the service to a previous (e.g., pre-release) state (e.g., when CIOS detects service health degradation while executing).

[0055] The CIOS can measure service health by monitoring alarms and executing integration tests. The CIOS can help teams quickly define roll-back behavior in the event of service degradation, which it can later execute. The CIOS can generate and display plans and can track approval. The CIOS can combine the functionality of provisioning and deployment in a single system that coordinates these tasks across a region build. The CIOS also supports the discovery of flocks (e.g., service resources such as flock config(s) corresponding to any suitable number of services), artifacts, resources, and dependencies. The CIOS can discover dependencies between execution tasks at every level (e.g., resource level, execution target level, phase level, service level, etc.) through a static analysis (e.g., including parsing and processing content) of one or more configuration files. Using these dependencies, the CIOS can generate various data structures from these dependencies that can be used to drive task execution (e.g., tasks regarding provisioning of infrastructure resources and deployment of artifacts across the region).

[0056] FIG. 1 is a block diagram of an environment 100 in which a Cloud Infrastructure Orchestration Service (CIOS) 102 may operate to dynamically provide bootstrap services in a region, according to at least one embodiment. CIOS 102 can include, but is not limited to, the following components: Real-time Regional Data Distributor (RRDD) 104, Multi-Flock Orchestrator (MFO) 106, CIOS Central 108, CIOS Regional 110, and Capabilities Service 112. Specific functionality of CIOS Central 108 and CIOS Regional 110 is provided in more detail in U.S. Application No. 17/016,754, entitled "Techniques for Deploying Infrastructure Resources with a Declarative Provisioning Tool," the entire contents of which are incorporated in its entirety for all purposes. In some embodiments, any suitable combination of the components of CIOS 102 may be provided as a service. In some embodiments, some portion of CIOS 102 may be deployed to a region (e.g., a data center represented by host region 103). In some embodiments, CIOS 102 may include any suitable number of cloud

services (not depicted in FIG. 1) discussed in further detail in U.S. Application No. 17/016,754 and below with respect to FIGS. 2 and 3.

[0057] Real-time Regional Data Distributor (RRDD) 104 may be configured to maintain and provide region data that identifies realms, regions, execution targets, and availability domains. In some cases, the region data may be in any suitable form (e.g., JSON format, data objects/containers, XML, etc.). Region data maintained by RRDD 104 may include any suitable number of subsets of data which can individually be referenceable by a corresponding identifier. By way of example, an identifier "all_regions" can be associated with a data structure (e.g., a list, a structure, an object, etc.) that includes a metadata for all defined regions. As another example, an identifier such as "realms" can be associated with a data structure that identifies metadata for a number of realms and a set of regions corresponding to each realm. In general, the region data may maintain any suitable attribute of one or more realm(s), region(s), availability domains (ADs), execution target(s) (ETs), and the like, such as identifiers, DNS suffixes, states (e.g., a state of a region), and the like. The RRDD 104 may be configured to manage region state as part of the region data. A region state may include any suitable information indicating a state of bootstrapping within a region. By way of example, some example region states can include "initial," "building," "production," "paused," or "deprecated." The "initial" state may indicate a region that has not yet been bootstrapped. A "building" state may indicate that bootstrapping of one or more flocks within the region has commenced. A "production" state may indicate that bootstrapping has been completed and the region is ready for validation. A "paused" state may indicate that CIOS Central 108 or CIOS Regional 110 has paused internal interactions with the regional stack, likely due to an operational issue. A "deprecated" state may indicate the region has been deprecated and is likely unavailable and/or will not be contacted again.

[0058] CIOS Central 108 is configured to provide any suitable number of user interfaces with which users (e.g., user 109) may interact with CIOS 102. By way of example, users can make changes to region data via a user interface provided by CIOS Central 108. CIOS Central 108 may additionally provide a variety of interfaces that enable users to: view changes made to flock configs and/or artifacts, generate and view plans, approve/reject plans, view status on plan execution (e.g., corresponding to tasks involving infrastructure provisioning, deployment, region build, and/or desired state of any suitable number of resources managed by CIOS 102. CIOS Central 108 may implement a control plane configured to manage any suitable number of CIOS Regional 110 instances. CIOS Central

108 can provide one or more user interfaces for presenting region data, enabling the user 109 to view and/or change region data. CIOS Central 108 can be configured to invoke the functionality of RRDD 104 via any suitable number of interfaces. Generally, CIOS Central 108 may be configured to manage region data, either directly or indirectly (e.g., via RRDD 104). CIOS Central 108 may be configured to compile flock configs to inject region data as variables within the flock configs.

[0059] Each instance of CIOS Regional 110 may correspond to a component or module configured to execute bootstrapping tasks that are associated with a single service of a region. CIOS Regional 110 can receive desired state data from CIOS Central 108. In some embodiments, desired state data may include a flock config that declares (e.g., via declarative statements) a desired state of resources associated with a service. CIOS Central 108 can maintain current state data indicating any suitable aspect of the current state of the resources associated with a service. In some embodiments, CIOS Regional 110 can identify, through a comparison of the desired state data and the current state data, that changes are needed to one or more resources. For example, CIOS Regional 110 can determine that one or more infrastructure components need to be provisioned, one or more artifacts deployed, or any suitable change needed to the resources of the service to bring the state of those resources in line with the desired state. As CIOS Regional 110 performs bootstrapping operations, it may publish data indicating various capabilities of a resource as they become available. A "capability" identifies a unit of functionality associated with a service. The unit could be a portion, or all of the functionality to be provided by the service. By way of example, a capability can be published indicating that a resource is available for authorization/authentication processing (e.g., a subset of the functionality to be provided by the resource). As another example, a capability can be published indicating the full functionality of the service is available. Capabilities can be used to identify functionality on which a resource or service depends and/or functionality of a resource or service that is available for use.

[0060] Capabilities Service 112 is configured to maintain capabilities data that indicates 1) what capabilities of various services are currently available, 2) whether any resource/service is waiting on a particular capability, 3) what particular resources and/or services are waiting on a given capability, or any suitable combination of the above. Capabilities Service 112 may provide an interface with which capabilities data may be requested. Capabilities Service 112 may provide one or more interfaces (e.g., application programming interfaces) that enable

Capabilities Service 112 to transmit capabilities data to MFO 106 and/or CIOS Regional 110 (e.g., each instance of CIOS Regional 110). In some embodiments, MFO 106 and/or any suitable component or module of CIOS Regional 110 may be configured to request capabilities data from Capabilities Service 112.

[0061] In some embodiments, Multi-Flock Orchestrator (MFO) 106 may be configured to drive region build efforts. In some embodiments, MFO 106 can manage information that describes what flock/flock config versions and/or artifact versions are to be utilized to bootstrap a given service within a region (or to make a unit of change to a target region). In some embodiments, MFO 106 may be configured to monitor (or be otherwise notified of) changes to the region data managed by Real-time Regional Data Distributor 104. In some embodiments, receiving an indication that region data has been changed may cause a region build to be triggered by MFO 106. In some embodiments, MFO 106 may collect various flock configs and artifacts to be used for a region build. Some, or all, of the flock configs may be configured to be region agnostic. That is, the flock configs may not explicitly identify what regions to which the flock is to be bootstrapped. In some embodiments, MFO 106 may trigger a data injection process through which the collected flock configs are recompiled (e.g., by CIOS Central 108). During recompilation, operations may be executed (e.g., by CIOS Central 108) to cause the region data maintained by Real-time Regional Data Distributor 104 to be injected into the config files. Flock configs can reference region data through variables/parameters without requiring hard-coded identification of region data. The flock configs can be dynamically modified at run time using this data injection rather than having the region data be hardcoded, and therefore, and more difficult to change.

[0062] Multi-Flock Orchestrator 106 can perform a static flock analysis in which the flock configs are parsed to identify dependencies between resources, execution targets, phases, and flocks, and in particular to identify circular dependencies that need to be removed. In some embodiments, MFO 106 can generate any suitable number of data structures based on the dependencies identified. These data structures (e.g., directed acyclic graph(s), linked lists, etc.) may be utilized by the Cloud Infrastructure Orchestration Service 102 to drive operations for performing a region build. By way of example, these data structures may collectively define an order by which services are bootstrapped within a region. An example of such a data structure is discussed further below with respect to Build Dependency Graph 338 of FIG. 3. If circular dependencies (e.g., service A requires service B and vice versa) exist and are identified through the static flock analysis and/or graph, MFO may be

configured to notify any suitable service teams that changes are required to the corresponding flock config to correct these circular dependencies. MFO 106 can be configured to traverse one or more data structures to manage an order by which services are bootstrapped to a region. MFO 106 can identify (e.g., using data obtained from Capabilities Service 112) capabilities available within a given region at any given time. MFO 106 can use this data to identify when CIOS Regional 110 can bootstrap a service, when bootstrapping is blocked, and/or when bootstrapping operations associated with a previously blocked service can resume. Based on this traversal, MFO 106 can perform a variety of releases in which instructions are transmitted by MFO 106 to CIOS Central 108 to perform bootstrapping operations corresponding to any suitable number of flock configs. In some examples, MFO 106 may be configured to identify that one or more flock configs may require multiple releases due to circular dependencies found within the graph. As a result, MFO 106 may transmit multiple instruction sets to CIOS Central 108 for a given flock config to break the circular dependencies identified in the graph.

[0063] In some embodiments, a user can request that a new region (e.g., target region 114) be built. This can involve bootstrapping resources corresponding to a variety of services. In some embodiments, target region 114 may not be communicatively available (and/or secure) at a time at which the region build request is initiated. Rather than delay bootstrapping until such time as target region 114 is available and configured to perform bootstrapping operations, CIOS 102 may initiate the region build using a virtual bootstrap environment 116. Virtual bootstrap environment (ViBE) 116 may be an overlay network that is hosted by host region 103 (a preexisting region that has previously been configured with a core set of services and which is communicatively available and secure). MFO 106 can leverage resources of the host region 103 to bootstrap resources to the ViBE 116 (generally referred to as "building the ViBE"). By way of example, MFO 106 can provide instructions through CIOS Central 108 that cause an instance of CIOS Regional 110 within a host region (e.g., host region 103) to bootstrap another instance of CIOS Regional within the ViBE 116. Once the CIOS Regional within the ViBE is available for processing, bootstrapping the services for the target region 114 can continue within the ViBE 116. When target region 114 is available to perform bootstrapping operations, the previously bootstrapped services within ViBE 116 may be migrated to target region 114. Utilizing these techniques, CIOS 102 can greatly improve the speed at which a region is built by drastically reducing the need for any manual input and/or configuration to be provided.

[0064] FIG. 2 is a block diagram for illustrating an environment 200 and method for building a virtual bootstrap environment (ViBE) 202 (an example of ViBE 116 of FIG. 1), according to at least one embodiment. ViBE 202 represents a virtual cloud network that is provisioned in the overlay of an existing region (e.g., host region 204, an example of the host region 103 of FIG. 1 and in an embodiment is a Host Region Service Enclave). ViBE 202 represents an environment in which services can be staged for a target region (e.g., a region under build such as target region 114 of FIG. 1) before the target region becomes available.

[0065] In order to bootstrap a new region (e.g., target region 114 of FIG. 1), a core set of services may be bootstrapped. While those core set of services exist in the host region 204, they do not yet exist in the ViBE (nor the target region). These essential core services provide the functionality needed to provision devices, establish a chain of trust to the new region, and deploy remaining services (e.g., flocks) into a region. The ViBE 202 may be a tenancy that is deployed in a host region 204. It can be thought of as a virtual region.

[0066] When the target region is available to provide bootstrapping operations, the ViBE 202 can be connected to the target region so that services in the ViBE can interact with the services and/or infrastructure components of the target region. This will enable deployment of production level services, instead of self-contained seed services as in previous systems, and will require connectivity over the internet to the target region. Conventionally, a seed service was deployed as part of a container collection and used to bootstrap dependencies necessary to build out the region. Using infrastructure/tooling of an existing region, resources may be bootstrapped (e.g., provisioned and deployed) into the ViBE 202 and connected to the service enclave of a region (e.g., host region 204) in order to provision hardware and deploy services until the target region is self-sufficient and can be communicated with directly. Utilizing the ViBE 202 allows for standing up the dependencies and services needed to be able to provision/prepare infrastructure and deploy software while making use of the host region's resources in order to break circular dependencies of core services.

[0067] Multi-Flock Orchestrator (MFO) 206 may be configured to perform operations to build (e.g., configure) ViBE 202. MFO 206 can obtain applicable flock configs corresponding to various resources to be bootstrapped to the new region (in this case, a ViBE region, ViBE 202). By way of example, MFO 206 may obtain a flock config (e.g., a "ViBE flock config") that identifies aspects of bootstrapping Capabilities Service 208 and Worker

210. As another example, MFO 206 may obtain another flock config corresponding to bootstrapping Domain Name Service (DNS) 212 to ViBE 202.

[0068] At step 1, MFO 206 may instruct CIOS Central 214 (e.g., an example of CIOS Central 108 and CIOS Central 214 of FIGS. 1 and 2, respectively). For example, MFO 206 may transmit a request (e.g., including the ViBE flock config) to request bootstrapping of the Capabilities Service 208 and Worker 210 that, at this time do not yet exist in the ViBE 202. In some embodiments, CIOS Central 214 may have access to all flock configs. Therefore, in some examples, MFO 206 may transmit an identifier for the ViBE flock config rather than the file itself, and CIOS Central 214 may independently obtain an identifier from storage (e.g., from DB 308 or flock DB 312 of FIG. 3).

[0069] At step 2, CIOS Central 214 may provide the ViBE flock config via a corresponding request to CIOS Regional 216. CIOS Regional 216 may parse the ViBE flock config to identify and execute specific infrastructure provisioning and deployment operations at step 3.

[0070] In some embodiments, the CIOS Regional 216 may utilize additional corresponding services for provisioning and deployment. For example, at step 4, CIOS Regional 216 CIOS Regional may instruct deployment orchestrator 218 (e.g., an example of a core service, or other write, build, and deploy applications software, of the host region 204) to execute instructions that in turn cause Capabilities Service 208 and Worker 210 to be bootstrapped within ViBE 202.

[0071] At step 5, a capability may be transmitted to the Capabilities Service 208 (from the CIOS Regional 216, Deployment Orchestrator 218 via the Worker 210 or otherwise) indicating that resources corresponding to the ViBE flock are available. Capabilities Service 208 may persist this data. In some embodiments, the Capabilities Service 208 adds this information to a list which maintains available capabilities with the ViBE. By way of example, the capability provided to Capabilities Service 208 at step 5 may indicate the Capabilities Service 208 and Worker 210 are available for processing.

[0072] At step 6, MFO 206 may identify that the capability indicating that Capabilities Service 208 and Worker 210 are available based on receiving or obtaining data (an identifier corresponding to the capability) from the Capabilities Service 208.

[0073] At step 7, as a result of receiving/obtaining the data at step 6, the MFO 206 may instruct CIOS Central 214 to bootstrap a DNS service (e.g., DNS 212) to the ViBE 202. The

instructions may identify or include a particular flock config corresponding to the DNS service.

[0074] At step 8, the CIOS Central 214 may instruct the CIOS Regional 216 to deploy DNS 212 to the ViBE 202. In some embodiments, the DNS flock config for the DNS 212 is provided by the CIOS Central 214.

[0075] At step 9, Worker 210, which is now deployed in the ViBE 202, may be assigned by CIOS Regional 216 to the task of deploying DNS 212. Worker 210 may execute a declarative infrastructure provisioner in the manner described above in connection with FIG. 3 to identify (e.g., from comparing the flock config (the desired state) to a current state of the (currently non-existing) resources associated with the flock) a set of operations that need to be executed to deploy DNS 212.

[0076] At step 10, the Deployment Orchestrator 218 may instruct Worker 210 to deploy DNS 212 in accordance with the operations identified at step 9. As depicted, Worker 210 proceeds with executing operations to deploy DNS 212 to ViBE 202 at step 11. At step 12, Worker 210 notifies Capabilities Service 208 that DNS 212 is available in ViBE 202. MFO 206 may subsequently identify that the resources associated with the ViBE flock config and the DNS flock config are available any may proceed to bootstrap any suitable number of additional resources to the ViBE.

[0077] After steps 1-12 are concluded, the process for building the ViBE 202 can be considered complete and the ViBE 202 can be considered built.

[0078] FIG. 3 is a block diagram for illustrating an environment 300 and method for bootstrapping services to a target region utilizing the ViBE, according to at least one embodiment.

[0079] At step 1, user 302 may utilize any suitable user interface provided by CIOS Central 304 (an example of CIOS Central 108 and CIOS Central 214 of FIG. 1 and 2, respectively) to modify region data. By way of example, user 302 may create a new region to which a number of services are to be bootstrapped.

[0080] At step 2, CIOS Central 304 may execute operations to send the change to RRDD 306 (e.g., an example of RRDD 104 of FIG. 1). At step 3, RRDD 306 may store the received region data in database 308, a data store configured to store region data including any suitable identifier, attribute, state, etc. of a region, AD, realm, ET, or the like. In some embodiments, updater 307 may be utilized to store region data in database 308 or any suitable data store

from which such updates may be accessible (e.g., to service teams). In some embodiments, updater 307 may be configured to notify (e.g., via any suitable electronic notification) of updates made to database 308.

[0081] At step 4, MFO 310 (an example of the MFO 106 and 206 of FIGS. 1 and 2, respectively) may detect the change in region data. In some embodiments, MFO 310 may be configured to poll RRDD 306 for changes in region data. In some embodiments, RRDD 306 may be configured to publish or otherwise notify MFO 310 of region changes.

[0082] At step 5, detecting the change in region data may trigger MFO 310 to obtain a version set (e.g., a version set associated with a particular identifier such as a "golden version set" identifier). identifying a particular version for each flock (e.g., service) that is to be bootstrapped to the new region and a particular version for each artifact corresponding to that flock. The version set may be obtained from DB 312. As flocks evolve and change, the versions for their corresponding configs and artifacts used for region build may change. These changes may be persisted in flock DB 312 such that MFO 310 may identify which versions of flock configs and artifacts to use for building a region (e.g., a ViBE region, a Target Region/non-ViBE Region, etc.). The flock configs (e.g., all versions of the flock configs) and/or artifacts (e.g., all versions of the artifacts) may be stored in DB 308, DB 312, or any suitable data store accessible to the CIOS Central 304 and/or MFO 310.

[0083] At step 6, MFO 310 may request CIOS Central 304 to recompile of each of the flock configs associated with the version set with the current region data. In some embodiments, the request may indicate a version for each flock config and/or artifact corresponding to those flock configs.

[0084] At step 7, CIOS Central 304 may obtain current region data from the DB 308 (e.g., directly, or via Real-time Regional Data Distributor 306) and retrieve any suitable flock config and artifact in accordance with the versions requested by MFO 310.

[0085] At step 8, CIOS Central 304 may recompile the flock configs with the region data obtained at step 7 to inject the flock configs with current region data. CIOS Central 304 may return the compiled flock configs to MFO 310. In some embodiments, CIOS Central 304 may simply indicate compilation is done, and MFO 310 may access the recompiled flock configs via RRDD 306.

[0086] At step 9, MFO 310 may perform a static analysis of the recompiled flock configs. As part of the static analysis, MFO 310 may parse the flock configs (e.g., using a library

associated with a declarative infrastructure provisioner (e.g., Terraform, or the like)) to identify dependencies between flocks. From the analysis and the dependencies identified, MFO 310 can generate Build Dependency Graph 338. Build Dependency Graph 338 may be an acyclic directed graph that identifies an order by which flocks are to be bootstrapped (and/or changes indicated in flock configs are to be applied) to the new region. Each node in the graph may correspond to bootstrapping any suitable portion of a particular flock. The specific bootstrapping order may be identified based at least in part on the dependencies. In some embodiments, the dependencies may be expressed as an attribute of the node and/or indicated via edges of the graph that connect the nodes. MFO 310 may traverse the graph (e.g., beginning at a starting node) to drive the operations of the region build.

[0087] In some embodiments, MFO 310 may utilize a cycle detection algorithm to detect the presence of a cycle (e.g., service A depends on service B and vice versa). MFO 310 can identify orphaned capabilities dependencies. For example, MFO 310 can identify orphaned nodes of the Build Dependency Graph 338 that do not connect to any other nodes. MFO 310 may identify falsely published capabilities (e.g., when a capability was prematurely published, and the corresponding functionality is not actually yet available). MFO 310 can detect from the graph that one or more instances of publishing the same capability exist. In some embodiments, any suitable number of these errors may be detected and MFO 310 (or another suitable component such as CIOS Central 304) may be configured to notify or otherwise present this information to users (e.g., via an electronic notification, a user interface, or the like). In some embodiments, MFO 310 may be configured to force delete/recreate resources to break circular dependencies and may once again provide instructions to CIOS Central 304 to perform bootstrapping operations for those resources and/or corresponding flock configs.

[0088] A starting node may correspond to bootstrapping the ViBE flock, a second node may correspond to bootstrapping DNS. The steps 10-15 correspond to deploying (via deployment orchestrator 317, an example of the deployment orchestrator 218 of FIG. 2) a ViBE flock to ViBE 316 (e.g., an example of ViBE 116 and 202 of FIGS. 1 and 2, respectively). That is, steps 10-15 of FIG. 3 generally correspond to steps 1-6 of FIG. 2. Once notified that capabilities exist corresponding to the ViBE flock being deployed (e.g., indicating that Capabilities Service 318 and Worker 320, corresponding to Capabilities Service 208 and Worker 210 of FIG. 2, are available) the MFO 310 recommences traversal of the Build Dependency Graph 338 to identify next operations to be executed.

[0089] By way of example, MFO 310 may continue traversing the Build Dependency Graph 338 to identify that a DNS flock is to be deployed. Steps 16-21 may be executed to deploy DNS 322 (an example of the DNS 212 of FIG. 2). These operations may generally correspond to steps 7-12 of FIG. 2.

[0090] At step 21, a capability may be stored indicating that DNS 322 is available. Upon detecting this capability, MFO 310 may recommence traversal of the Build Dependency Graph 338. On this traversal, the MFO 310 may identify that any suitable portion of an instance of CIOS Regional (e.g., an example of CIOS Regional 314) is to be deployed to the ViBE 316. In some embodiments, steps 16-21 may be substantially repeated with respect to deploying CIOS Regional (ViBE) 326 (an instance of CIOS Regional 314, CIOS Regional 110 of FIG. 1) and Worker 328 to the ViBE 316. A capability may be transmitted to the Capabilities Service 318 that CIOS Regional (ViBE) 326 is available.

[0091] Upon detecting the CIOS Regional (ViBE) 326 is available, MFO 310 may recommence traversal of the Build Dependency Graph 338. On this traversal, the MFO 310 may identify that a deployment orchestrator (e.g., Deployment Orchestrator 330, an example of the Deployment Orchestrator 317) is to be deployed to the ViBE 316. In some embodiments, steps 16-21 may be substantially repeated with respect to deploying Deployment Orchestrator 330. Information that identifies a capability may be transmitted to the Capabilities Service 318, indicating that Deployment Orchestrator 330 is available.

[0092] After Deployment Orchestrator 330 is deployed, ViBE 316 may be considered available for processing subsequent requests. Upon detecting Deployment Orchestrator 330 is available, MFO 310 may instruct subsequent bootstrapping requests to be routed to ViBE components rather than utilizing host region components (components of host region 332). Thus, MFO 310 can continue traversing the Build Dependency Graph 338, at each node instructing flock deployment to the ViBE 316 via CIOS Central 304. CIOS Central 304 may request CIOS Regional (ViBE) 326 to deploy resources according to the flock config.

[0093] At some point during this process, Target Region 334 may become available. Indication that the Target Region is available may be identifiable from region data for the Target Region 334 being provided by the user 302 (e.g., as an update to the region data). The availability of Target Region 334 may depend on establishing a network connection between the Target Region 334 and external networks (e.g., the Internet). The network connection may be supported over a public network (e.g., the Internet), but use software security tools

(e.g., IPSec) to provide one or more encrypted tunnels (e.g., IPSec tunnels such as tunnel 336) from the ViBE 316 to Target Region 334. As used herein, "IPSec" refers to a protocol suite for authenticating and encrypting network traffic over a network that uses Internet Protocol (IP), and can include one or more available implementations of the protocol suite (e.g., Openswan, Libreswan, strongSwan, etc.). The network may connect the ViBE 316 to the service enclave of the Target Region 334.

[0094] Prior to establishing the IPSec tunnels, the initial network connection to the Target Region 334 may be on a connection (e.g., an out-of-band VPN tunnel) sufficient to allow bootstrapping of networking services until an IPSec gateway may be deployed on an asset (e.g., bare-metal asset) in the Target Region 334. To bootstrap the Target Region's 334 network resources, Deployment Orchestrator 330 can deploy the IPSec gateway at the asset within Target Region 334. The Deployment Orchestrator 330 may then deploy VPN hosts at the Target Region 334 configured to terminate IPSec tunnels from the ViBE 316. Once services (e.g., Deployment Orchestrator 330, Service A, etc.) in the ViBE 316 can establish an IPSec connection with the VPN hosts in the Target Region 334, bootstrapping operations from the ViBE 316 to the Target Region 334 may begin.

[0095] In some embodiments, the bootstrapping operations may begin with services in the ViBE 316 provisioning resources in the Target Region 334 to support hosting instances of core services as they are deployed from the ViBE 316. For example, a host provisioning service may provision hypervisors on infrastructure (e.g., bare-metal hosts) in the Target Region 334 to allocate computing resources for VMs. When the host provisioning service completes allocation of physical resources in the Target Region 334, the host provisioning service may publish information indicating a capability that indicates that the physical resources in the Target Region 334 have been allocated. The capability may be published to Capabilities Service 318 via CIOS Regional (ViBE) 326 (e.g., by Worker 328).

[0096] With the hardware allocation of the Target Region 334 established and posted to capabilities service 318, CIOS Regional (ViBE) 326 can orchestrate the deployment of instances of core services from the ViBE 316 to the Target Region 334. This deployment may be similar to the processes described above for building the ViBE 316, but using components of the ViBE (e.g., CIOS Regional (ViBE) 326, Worker 328, Deployment Orchestrator 330) instead of components of the Host Region 332 service enclave. The deployment operations may generally correspond to steps 16-21 described above.

[0097] As a service is deployed from the ViBE 316 to the Target Region 334, the DNS record associated with that service may correspond to the instance of the service in the ViBE 316. The DNS record associated with the service may be updated at a later time to complete deployment of the service to the Target Region 334. Said another way, the instance of the service in the ViBE 316 may continue to receive traffic (e.g., requests) to the service until the DNS record is updated. A service may deploy partially into the Target Region 334 and publish information indicating a capability (e.g., to Capabilities Service 318) that the service is partially deployed. For example, a service running in the ViBE 316 may be deployed into the Target Region 334 with a corresponding compute instance, load balancer, and associated applications and other software, but may need to wait for database data to migrate to the Target Region 334 before being completely deployed. The DNS record (e.g., managed by DNS 322) may still be associated with the service in the ViBE 316. Once data migration for the service is complete, the DNS record may be updated to point to the operational service deployed in the Target Region 334. The deployed service in the Target Region 334 may then receive traffic (e.g., requests) for the service, while the instance of the service in the ViBE 316 may no longer receive traffic for the service.

Migrating Services from a ViBE to a Target Region

[0098] As discussed above, a CSP may build or deploy data centers to provide cloud services to its customers in new regions. The regions may correspond to a general geographic area, which may be preferably in proximity to new customers or customers with expanding service needs (e.g., customer growth in requiring scale-up of cloud services, expansion of cloud services to a new geographic area, etc.). To support building of new regions, a CIOS (e.g., CIOS 102 of FIG. 1) may be used to generate a ViBE (e.g., ViBE 316 of FIG. 3) in a designated host region (e.g., a one or more data centers in a host region).

[0099] The ViBE may be created in advance or in parallel with the building of the data centers in the new region. For example, because the building and provisioning of physical components (e.g., bare metal hosts, racks, networking switches, etc.) can take several months, a ViBE may be built in advance (e.g., days, weeks, etc.) of the completion of the new data center. Operators may deploy core services to the ViBE to support the deployment of services to the new data center once the physical infrastructure of the new data center is ready to host cloud services. By creating the ViBE in advance of (or in parallel with) the build of a region's

data center, the core services deployed therein may be evaluated for deployment readiness (e.g., production testing, resolving dependencies, etc.).

[0100] Provisioning infrastructure and deploying services from the ViBE to the target region can include a "scale-out" process by which services in the ViBE (e.g., CIOS Regional (ViBE) 326, Deployment Orchestrator 330, etc.) deploy instances of ViBE core services to the target region. In the scale-out process, the core services are deployed to the target region in a similar manner as to how the services were deployed into the ViBE.

[0101] While the services are operating in the ViBE, they may create new resources (e.g., new data, new artifacts). Since these resources were created independently of the deployment of the service using CIOS, the CIOS may not be aware of the new resources when deploying the service to the target region (e.g., the flock configs for deployment to the target region may not have information for resources created in the ViBE by the services). To complete the scale-out process, ViBE resources may need to be discovered and migrated to the target region. Until the resources are in the target region, the services deployed in the target region may not be able to provide service in response to requests. To prevent traffic from being sent to the service in the target region until the service's data and other resources are fully migrated, the DNS record for the target service may not be updated during the initial scale-out of the service to the target region. The CIOS may later identify the DNS record for the target region service as a resource and update the DNS record in a DNS (e.g., a ViBE DNS, a target region DNS). Once the DNS record is updated to point to the service in the target region, the migration and scale-out process for the service is complete.

[0102] FIG. 4 is a block diagram of an environment 400 in which the Cloud Infrastructure Orchestration Service (CIOS) may utilize a Resource Hunter to discover resources during region build, according to at least one embodiment. Environment 400 may be an example of environment 100 of FIG. 1. The Cloud Infrastructure Orchestration Service (CIOS) 402 may be an example of the CIOS 102 of FIG. 1. Real-time Regional Data Distributor (RRDD) 404 may be an example of the Real-time Regional Data Distributor 104 and/or 306 of FIGS. 1 and 3, respectively. Multi-Flock Orchestrator (MFO) 406 may be an example of the Multi-Flock Orchestrator 104, 206, and/or 310 of FIGS. 1-3, respectively. CIOS Central 408 may be an example of the CIOS Central 108, 214, and/or 304 of FIGS. 1-3, respectively. CIOS Regional 410 may be an example of the CIOS Regional 110, 216, and/or 314 of FIGS. 1-3, respectively. Capabilities Service 412 may be an example of the Capabilities Service 112,

208, and/or 318 of FIGS. 1-3, respectively. Target Region 414 may be an example of the Target Region 114 and/or 334 of FIGS. 1 and 3, respectively. Virtual Bootstrap Environment 416 may be an example of the Virtual Bootstrap Environment 114, 202, and/or 316 of FIGS. 1-3, respectively. The components of CIOS 402, including RRDD 404, MFO 406, CIOS Central 408, CIOS Regional 410, and Capabilities Service 412, may each perform the respective functionality discussed above in FIGS. 1-3 in connection with the corresponding components of FIGS. 1-3.

[0103] Environment 400 may include Resource Hunter Service (RHS) 420. RHS 420 may be configured to attempt discovery of resources at any suitable time. By way of example, any suitable number of resources may exist in any suitable region prior to a region build of that, or another region. By way of example, resource(s) 422 may include any suitable number of resources (e.g., infrastructure resources, artifacts, configuration files, etc.) that exist in the host region 403 (an example of the host region 103, 204, and/or 332 of FIGS. 1-3, respectively). Resource(s) 424 may exist in the ViBE 416 and/or resource(s) 426 may exist in a target region. Any suitable combination of resource(s) 422-426 may be created at any suitable time prior to, during, or after a region build process is executed to bootstrap any suitable number of services within ViBE 416 and/or Target Region 414.

[0104] RHS 420 may be configured to receive flock config information from MFO 406 (and/or any suitable component of the CIOS 402 and/or any service bootstrapped within the Host Region 403, the ViBE 416, and/or the Target Region 414). In some embodiments, MFO 406 may provide an identifier of the flock and RHS 420 may be configured to access corresponding flock config files associated with that flock. Alternatively, MFO 406 may provide the flock config to RHS 420. An example flock config is discussed in more detail in connection with FIG. 5 below.

[0105] RHS 420 may identify resource discovery data within the flock config. "Resource discovery data" refers to any suitable data (e.g., a set of parameters) with which a previously existing resource (e.g., one or more of resource(s) 422-426) may be identified. RHS 420 may execute operations to identify any suitable number of the resource(s) 422-426 utilizing the set of parameters of the resource discovery data. For example, RHS 420 may search (e.g., within a region specified by the resource discovery data) for particular resources that are associated with attributes that match the set of parameters provided in the resource discovery data. If one or more matching resources are discovered, RHS 420 may provide the identifier(s) for the

matching resource(s) to MFO 406 directly, and/or RHS 420 may store the identifier(s) within a record accessible to MFO 406.

[0106] MFO 406 may include a State Manager 428. The State Manager 428 may be configured to implement a state machine that transitions between states to drive a pass. By way of example, a single flock may require multiple releases (e.g., multiple transmissions of instructions to CIOS Central 408, multiple releases corresponding to one or more flock configs associated with a single service, etc.). State Manager 428 may be configured to coordinate the operations performed for a single release. This may include monitoring for messages from CIOS Central 408 and/or Resource Hunter Service 420. In some embodiments, the State Manager 428 may be configured to monitor one or more capabilities transmitted by Capabilities Service 412.

[0107] FIG. 5 is a block diagram depicting an example flow for executing operations for provisioning and deploying services to a data center of a new region and importing resources before updating a DNS record for the deployed service, according to at least one embodiment.

[0108] FIG. 5 is a block diagram depicting an environment 500 and an example method for executing operations for provisioning and deploying services to a data center of a new region and importing resources before updating another service with the resources, according to at least one embodiment. Operations described herein with respect to FIG. 5 may be considered an extension of the method described above with respect to FIG. 3. For example, MFO 510 (an example of MFO 310 of FIG. 3) may deploy a service (e.g., service 502) into the target region 534 by instructing CIOS Central 504 to bootstrap the service into the target region similarly to how services (e.g., Deployment Orchestrator 530, an example of Deployment Orchestrator 330 of FIG. 3) were deployed into the ViBE. In addition, one or more of the operations described herein with respect to FIG. 5 may be similar to operations described above with respect to FIG. 4. For example, the environment 500 can include a Resource Hunter Service (RHS) 511 that may be similar to RHS 420 of FIG. 4.

[0109] The operations described with respect to FIG. 5 (and any other method or process described herein) may be implemented in hardware, computer instructions, or a combination thereof. In the context of computer instructions, the operations represent computer-executable instructions stored on one or more computer-readable storage media that, when executed by one or more processors, perform the recited operations. Generally, computer-executable

instructions include routines, programs, objects, components, data structures, and the like that perform particular functions or implement particular data types. The order in which the operations are described is not intended to be construed as a limitation, and any number of the described operations can be omitted or combined in any order and/or in parallel to implement the processes.

[0110] The migration of services in a virtual bootstrap environment (ViBE) may be referred to herein as a "scale-out" process, and may include operations typically referred to as "migration" within the context of distributed computing environments (e.g., migrating databases, migrating application data, etc.). The environment 500 can include a Host Region 532 in which a ViBE 516 has been implemented. The environment 500 can also include a Target Region 534 into which services hosted in the ViBE 516 may be scaled-out. The Host Region 532 and the Target Region 534 may be examples of the Host Region 332 and Target Region 334 of FIG. 3, respectively.

[0111] Prior to beginning a scale-out process from the ViBE 516, a network connection may be established between the ViBE 516 in the host region 532 and the target region 534. The network connection can include one or more tunnels 536 (an example of tunnels 336 of FIG. 3). The tunnels 536 may be encrypted tunnels (e.g., IPSec tunnels). The network connection may form one or more virtual private network (VPN) connections between a network hosting the ViBE 516 (e.g., a virtual cloud network (VCN) of host region 532) and a service enclave (e.g., a VCN of Target Region 534) of the Target Region 534.

[0112] Steps 1-8 may be executed to deploy Service A 502 into the target region 534. The operations of steps 1, 2, 3, and 8 may generally correspond to steps 7, 8, 9, and 12 of FIG. 2. Steps 4-5 include operations related to provisioning infrastructure resources at the Target Region 534, while steps 6-8 may be similar to an iteration of steps 10 and 11 of FIG. 2 to deploy a service into the Target Region 534.

[0113] The steps described below include infrastructure provisioning (e.g., steps 4-5) as a portion of a method using CIOS to provision infrastructure resources as a preliminary part of deploying a service (e.g., a service according to a flock config using a declarative provisioning tool). In some embodiments, provisioning infrastructure resources (e.g., infrastructure resources 503) may be performed as a separate operation under the direction of CIOS Central 504 and MFO 510. For example, a host provisioning service (e.g., host provisioning service 506) may be deployed to the ViBE 516 and capable of provisioning

infrastructure resources in the Target Region 534 before a deployment orchestration service (e.g., Deployment Orchestrator 530) is fully deployed in the ViBE 516 to support deployment of services to the Target Region 534. In this instance, based on the capabilities provided to Capabilities Service 518, MFO 510 may begin provisioning infrastructure resources at the Target Region 534 using host provisioning service 506 without an accompanying deployment of a service.

[0114] At step 1, the MFO 510 may instruct CIOS Central 504 to deploy Service A 502 to the Target Region 534. Service A 502 may require infrastructure resources 503 to be provisioned at the Target Region 534. For example, Service A 502 may be executed with one or more compute instances executing on one or more VMs in the target region. The instructions to deploy Service A 502 may identify or include resources (e.g., infrastructure resources 503) to provision in the target region 534. Provisioning resources can include configuring one or more hosts (e.g., one or more virtual machines (VMs)) in a computing environment of the Target Region 534. The instructions to deploy service 502 may also identify or include a flock config corresponding to the Service A 502.

[0115] Service A 502 may be an instance of Service A 509 deployed in the ViBE 516. Since the ViBE 516 and Target Region 534 may be connected by Tunnels 536 to form a distributed virtual private network (VPN) between the ViBE 516 and the service enclave of the Target Region 534, the deployment of Service A 503 can result in duplicate service stacks available within the distributed VPN. To avoid conflicts and ensure that traffic is handled by only one instance of the service, the DNS record (e.g., the DNS managed by DNS 522) for the service may initially point to Service A 509 in the ViBE. The DNS record may be created as part of the deployment of Service A 509 into the ViBE (e.g., deployed using MFO 510 and CIOS Central 504). When Service A 502 is deployed to the Target Region 534, the DNS record associated with Service A may not be updated.

[0116] At step 2, CIOS Central 504 may instruct CIOS Regional (ViBE) 526 to provision infrastructure resources 503 and deploy Service A 502 to the Target Region 534. In some embodiments, the flock config for Service A 502 is provided by the CIOS Central 504.

[0117] At step 3, Worker 528 may be assigned by CIOS Regional (ViBE) 526 to the task of provisioning infrastructure resources 503 and deploying Service A 502. Worker 528 may execute a declarative provisioner to identify (e.g., by comparing the flock config to a current

state of the resources associated with the flock) a set of operations that need to be executed to deploy Service A 502.

[0118] At step 4, the Worker 528 may instruct the host provisioning service 506 to provision infrastructure resources 503 in the Target Region 534 in accordance with the operations identified at step 3, which may be done by host provisioning service 506 at step 5. At steps 6 and 7, the Worker 528 may instruct a compute control plane 508 and Deployment Orchestrator 530 to launch instances and deploy Service A 502 to the launched instances, respectively. The Compute Control Plane 508 and Deployment orchestrator 530 may do the instructed launching and deployment tasks at step 8.

[0119] At step 9, the Worker 528 may post one or more capabilities to Capabilities Service 518 that Service A 502 is available. The capability may indicate that the service has been "partially" deployed. For example, the capability may be a value like "servicea_partial_scaleout." As described above, Service A 502 may be successfully deployed onto the infrastructure resources 503 in the Target Region 534, but may not have necessary resources (e.g., service data, data resources, etc.) to handle service traffic. For example, Service A 509 in the ViBE 516 may create, update, store, or otherwise produce and/or operate with data in the ViBE 516. This data may include Resource 540. In some embodiments, Service A 540 may use a service (e.g., Data Storage Service 544) in the Host Region 532 service enclave to persist data associated with Service A 509 operations. This data can include Service Data 546. The data of Resource 540 and Service Data 546 may be generated by Service A 509 after deployment into the ViBE 516. Because the data is generated after deployment, MFO 510 and other components of CIOS (e.g., CIOS Central 504, CIOS Regional (ViBE) 526, etc.) may not be aware of the resource as part of the configuration (e.g., flock config) for deploying the instance of Service A 502 to the Target Region.

[0120] MFO 510 may obtain the published partial scale-out capability. At step 10, MFO 510 can provide an identifier of the Service A 502 that indicated a partial scale-out (e.g., an identifier for the flock corresponding to Service A 502) to RHS 511. RHS 511 may identify resource discovery data within a flock config corresponding to Service A 502 and use the resource discovery data to identify resources (e.g., Resource 540). The RHS 511 can provide the resource (or a resource identifier) to MFO 510.

[0121] At step 11, MFO 510 can use the resource (or resource identifier) provided by RHS 511 to configure another pass for deploying the flock corresponding to Service A 502 (e.g., another pass for deploying and/or updating Service A 502). The second pass can include operations similar to steps 1-3 to instruct a worker (e.g., Worker 528) to execute operations in conjunction with one or more services in the ViBE 516 to deploy an update to Service A 502. The update can include updating a service in the ViBE (e.g., DNS 522) or a service in the Target Region 534 (e.g., a DNS scaled-out into the Target Region 534).

[0122] For the example where the resource is a DNS record, at step 12 CIOS Central 504 may instruct CIOS Regional (ViBE) 526 to deploy Service A 502 according to a flock config updated using the discovered DNS record.

[0123] At step 13, Worker 528 may be assigned by CIOS Regional (ViBE) 526 to the task of updating the DNS record. Worker 528 may execute a declarative provisioner to identify (e.g., by comparing the flock config to a current state of the resources associated with the flock) a set of operations that need to be executed to deploy Service A 502. Since the updated DNS record may be the only difference between the current state of the deployed flock corresponding to Service A 502, the declarative provisioner may only produce instructions to update the DNS record (rather than redeploy all resources of Service A 502). The Worker 528 may update DNS 522 with the new DNS record. The DNS record update may result in the DNS 522 identifying Service A 502 as the destination for Service A traffic within the VPN between the ViBE 516 and the Target Region 534.

[0124] In some embodiments, the resource identified by RHS 511 can include data created by Service A 509 (e.g., Resource 540). MFO 510, CIOS Central 504, and CIOS Regional (ViBE) 526 may perform similar operations as steps 1-3 and 11-13 to deploy Resource 540 from the ViBE 516 to the Target Region 542. In this way, Service A 502 may be provided with Resource 542, which can include the data of Resource 540.

[0125] Prior to performing the CIOS pass that updates the DNS record, resources may be migrated to the Target Region 534. Resources associated with services in the ViBE 516 that are in the Host Region 532 but outside of the ViBE 516 (e.g., resources in the service enclave of the Host Region 532) may be migrated to the Target Region 534 as part of a database migration process or other data transfer process that operates independently of the operations of the CIOS (indicated as step 15 in FIG. 5). In some examples, the resources that are migrated to the Target Region 534 can include secrets (e.g., credentials, passwords, keys,

tokens, etc.) used for authentication and/or authorization. To improve security, secrets that are migrated to the Target Region 534 may be re-encrypted using a new master key associated with the Target Region 534. The migrated secrets may then be rotated (e.g., changed) within the Target Region 534 to prevent any secrets that persist in the Host Region 532 from being usable with the Target Region 534.

[0126] It should be appreciated that the above description of operations for scaling out Service A 502 into the Target Region 534 apply similarly to any service hosted in the ViBE 516. Services hosted in the ViBE 516 are deployed to the Target Region 534 using CIOS, resources associated with those deployed services not originally deployed by CIOS are discovered by RHS (e.g., RHS 511), and service traffic is switched from being routed to the service in the ViBE to being routed to the service in the target region by updating the corresponding DNS record, which may occur after all necessary resources for the service are migrated from the ViBE, Host Region 532 service enclave, and/or other location.

[0127] FIG. 6 illustrates an example method 600 for migrating resources for services in a virtual bootstrap environment (ViBE), according to at least one embodiment. The method 600 may be performed by one or more components of a distributed computing system (e.g., a cloud computing system), including one or more components of the Cloud Infrastructure Orchestration Service 102 of FIG. 1. A computer-readable storage medium may comprise computer-readable instructions that, upon execution by one or more processors of a distributed computing system, cause the distributed computing system to perform the method 600. The operations of method 600 may be performed in any suitable order, and method 600 may include more or fewer operations than those depicted in FIG. 6.

[0128] Some or all of the method 600 (or any other processes and/or methods described herein, or variations, and/or combinations thereof) may be performed under the control of one or more computer systems configured with executable instructions and may be implemented as code (e.g., executable instructions, one or more computer programs, or one or more applications) executing collectively on one or more processors, by hardware or combinations thereof. The code may be stored on a computer-readable storage medium, for example, in the form of a computer program comprising a plurality of instructions executable by one or more processors. The computer-readable storage medium may be non-transitory.

[0129] The method 600 may begin at block 602 when a virtual cloud network (VCN) may be generated in a data center of a host region. The VCN may be a Virtual Bootstrap Environment (ViBE) VCN.

[0130] At block 604, a virtual bootstrap environment (ViBE) (e.g., ViBE 516 of FIG. 5) may be implemented in the VCN. As described above, the ViBE may be configured to stage one or more services for deployment to a target region. The ViBE can host a plurality of services (e.g., Service A 509 of FIG. 5).

[0131] At block 606, an instance of one of the plurality of services in the ViBE may be deployed to a target region data center (e.g., a data center of Target Region 534 of FIG. 5). Deployment of the first service may be by a Multi Flock Orchestrator (MFO) (e.g., MFO 106 of FIG. 1) of the CIOS (e.g., CIOS 102 of FIG. 1). The MFO may leverage one or more services in the ViBE (e.g., CIOS Regional (ViBE) 526, Deployment Orchestrator 530, etc.) to deploy the instance of the service into the target region data center.

[0132] At block 608, an indication may be received from the deployed instance that the service was deployed successfully. The indication may be a capability published to a capabilities service (e.g., Capabilities Service 518 of FIG. 5). In some embodiments, the capability may indicate that the instance of the service was successfully partially deployed.

[0133] At block 610, a resource associated with the service may be identified. A Resource Hunting Service (e.g., RHS 511 of FIG. 5) may identify the resource. In some embodiments, the resource may be a DNS record for the instance in the target region data center.

[0134] At block 612, a second service of the plurality of services may be updated with the resource. In some embodiments, the second service is a DNS of the ViBE. In some embodiments, the service deployed to the target region may be updated with the resource.

Example Architectures for Providing Infrastructure-as-a-Service (IaaS)

[0135] As noted above, infrastructure as a service (IaaS) is one particular type of cloud computing. IaaS can be configured to provide virtualized computing resources over a public network (e.g., the Internet). In an IaaS model, a cloud computing provider can host the infrastructure components (e.g., servers, storage devices, network nodes (e.g., hardware), deployment software, platform virtualization (e.g., a hypervisor layer), or the like). In some cases, an IaaS provider may also supply a variety of services to accompany those infrastructure components (e.g., billing, monitoring, logging, load balancing and

clustering, etc.). Thus, as these services may be policy-driven, IaaS users may be able to implement policies to drive load balancing to maintain application availability and performance.

[0136] In some instances, IaaS customers may access resources and services through a wide area network (WAN), such as the Internet, and can use the cloud provider's services to install the remaining elements of an application stack. For example, the user can log in to the IaaS platform to create virtual machines (VMs), install operating systems (OSs) on each VM, deploy middleware such as databases, create storage buckets for workloads and backups, and even install enterprise software into that VM. Customers can then use the provider's services to perform various functions, including balancing network traffic, troubleshooting application issues, monitoring performance, managing disaster recovery, etc.

[0137] In most cases, a cloud computing model may require the participation of a cloud provider. The cloud provider may, but need not be, a third-party service that specializes in providing (e.g., offering, renting, selling) IaaS. An entity might also opt to deploy a private cloud, becoming its own provider of infrastructure services.

[0138] In some examples, IaaS deployment is the process of putting a new application, or a new version of an application, onto a prepared application server or the like. It may also include the process of preparing the server (e.g., installing libraries, daemons, etc.). This is often managed by the cloud provider, below the hypervisor layer (e.g., the servers, storage, network hardware, and virtualization). Thus, the customer may be responsible for handling (OS), middleware, and/or application deployment (e.g., on self-service virtual machines (e.g., that can be spun up on demand) or the like.

[0139] In some examples, IaaS provisioning may refer to acquiring computers or virtual hosts for use, and even installing needed libraries or services on them. In most cases, deployment does not include provisioning, and the provisioning may need to be performed first.

[0140] In some cases, there are two different challenges for IaaS provisioning. First, there is the initial challenge of provisioning the initial set of infrastructure before anything is running. Second, there is the challenge of evolving the existing infrastructure

(e.g., adding new services, changing services, removing services, etc.) once everything has been provisioned. In some cases, these two challenges may be addressed by enabling the configuration of the infrastructure to be defined declaratively. In other words, the infrastructure (e.g., what components are needed and how they interact) can be defined by one or more configuration files. Thus, the overall topology of the infrastructure (e.g., what resources depend on which, and how they each work together) can be described declaratively. In some instances, once the topology is defined, a workflow can be generated that creates and/or manages the different components described in the configuration files.

[0141] In some examples, an infrastructure may have many interconnected elements. For example, there may be one or more virtual private clouds (VPCs) (e.g., a potentially on-demand pool of configurable and/or shared computing resources), also known as a core network. In some examples, there may also be one or more inbound/outbound traffic group rules provisioned to define how the inbound and/or outbound traffic of the network will be set up and one or more virtual machines (VMs). Other infrastructure elements may also be provisioned, such as a load balancer, a database, or the like. As more and more infrastructure elements are desired and/or added, the infrastructure may incrementally evolve.

[0142] In some instances, continuous deployment techniques may be employed to enable deployment of infrastructure code across various virtual computing environments. Additionally, the described techniques can enable infrastructure management within these environments. In some examples, service teams can write code that is desired to be deployed to one or more, but often many, different production environments (e.g., across various different geographic locations, sometimes spanning the entire world). However, in some examples, the infrastructure on which the code will be deployed may need to first be set up. In some instances, the provisioning can be done manually, a provisioning tool may be utilized to provision the resources, and/or deployment tools may be utilized to deploy the code once the infrastructure is provisioned.

[0143] FIG. 7 is a block diagram 700 illustrating an example pattern of an IaaS architecture, according to at least one embodiment. Service operators 702 can be communicatively coupled to a secure host tenancy 704 that can include a virtual cloud

network (VCN) 706 and a secure host subnet 708. In some examples, the service operators 702 may be using one or more client computing devices, which may be portable handheld devices (e.g., an iPhone®, cellular telephone, an iPad®, computing tablet, a personal digital assistant (PDA)) or wearable devices (e.g., a Google Glass® head mounted display), running software such as Microsoft Windows Mobile®, and/or a variety of mobile operating systems such as iOS, Windows Phone, Android, BlackBerry 8, Palm OS, and the like, and being Internet, e-mail, short message service (SMS), Blackberry®, or other communication protocol enabled. Alternatively, the client computing devices can be general purpose personal computers including, by way of example, personal computers and/or laptop computers running various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems. The client computing devices can be 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, such as for example, Google Chrome OS. Alternatively, or in addition, client computing devices may be any other electronic device, such as a thin-client computer, an Internet-enabled gaming system (e.g., a Microsoft Xbox gaming console with or without a Kinect® gesture input device), and/or a personal messaging device, capable of communicating over a network that can access the VCN 706 and/or the Internet.

[0144] The VCN 706 can include a local peering gateway (LPG) 710 that can be communicatively coupled to a secure shell (SSH) VCN 712 via an LPG 710 contained in the SSH VCN 712. The SSH VCN 712 can include an SSH subnet 714, and the SSH VCN 712 can be communicatively coupled to a control plane VCN 716 via the LPG 710 contained in the control plane VCN 716. Also, the SSH VCN 712 can be communicatively coupled to a data plane VCN 718 via an LPG 710. The control plane VCN 716 and the data plane VCN 718 can be contained in a service tenancy 719 that can be owned and/or operated by the IaaS provider.

[0145] The control plane VCN 716 can include a control plane demilitarized zone (DMZ) tier 720 that acts as a perimeter network (e.g., portions of a corporate network between the corporate intranet and external networks). The DMZ-based servers may have restricted responsibilities and help keep breaches contained. Additionally, the DMZ tier

720 can include one or more load balancer (LB) subnet(s) 722, a control plane app tier 724 that can include app subnet(s) 726, a control plane data tier 728 that can include database (DB) subnet(s) 730 (e.g., frontend DB subnet(s) and/or backend DB subnet(s)). The LB subnet(s) 722 contained in the control plane DMZ tier 720 can be communicatively coupled to the app subnet(s) 726 contained in the control plane app tier 724 and an Internet gateway 734 that can be contained in the control plane VCN 716, and the app subnet(s) 726 can be communicatively coupled to the DB subnet(s) 730 contained in the control plane data tier 728 and a service gateway 736 and a network address translation (NAT) gateway 738. The control plane VCN 716 can include the service gateway 736 and the NAT gateway 738.

[0146] The control plane VCN 716 can include a data plane mirror app tier 740 that can include app subnet(s) 726. The app subnet(s) 726 contained in the data plane mirror app tier 740 can include a virtual network interface controller (VNIC) 742 that can execute a compute instance 744. The compute instance 744 can communicatively couple the app subnet(s) 726 of the data plane mirror app tier 740 to app subnet(s) 726 that can be contained in a data plane app tier 746.

[0147] The data plane VCN 718 can include the data plane app tier 746, a data plane DMZ tier 748, and a data plane data tier 750. The data plane DMZ tier 748 can include LB subnet(s) 722 that can be communicatively coupled to the app subnet(s) 726 of the data plane app tier 746 and the Internet gateway 734 of the data plane VCN 718. The app subnet(s) 726 can be communicatively coupled to the service gateway 736 of the data plane VCN 718 and the NAT gateway 738 of the data plane VCN 718. The data plane data tier 750 can also include the DB subnet(s) 730 that can be communicatively coupled to the app subnet(s) 726 of the data plane app tier 746.

[0148] The Internet gateway 734 of the control plane VCN 716 and of the data plane VCN 718 can be communicatively coupled to a metadata management service 752 that can be communicatively coupled to public Internet 754. Public Internet 754 can be communicatively coupled to the NAT gateway 738 of the control plane VCN 716 and of the data plane VCN 718. The service gateway 736 of the control plane VCN 716 and of the data plane VCN 718 can be communicatively couple to cloud services 756.

[0149] In some examples, the service gateway 736 of the control plane VCN 716 or of the data plane VCN 718 can make application programming interface (API) calls to cloud services 756 without going through public Internet 754. The API calls to cloud services 756 from the service gateway 736 can be one-way: the service gateway 736 can make API calls to cloud services 756, and cloud services 756 can send requested data to the service gateway 736. However, cloud services 756 may not initiate API calls to the service gateway 736.

[0150] In some examples, the secure host tenancy 704 can be directly connected to the service tenancy 719, which may be otherwise isolated. The secure host subnet 708 can communicate with the SSH subnet 714 through an LPG 710 that may enable two-way communication over an otherwise isolated system. Connecting the secure host subnet 708 to the SSH subnet 714 may give the secure host subnet 708 access to other entities within the service tenancy 719.

[0151] The control plane VCN 716 may allow users of the service tenancy 719 to set up or otherwise provision desired resources. Desired resources provisioned in the control plane VCN 716 may be deployed or otherwise used in the data plane VCN 718. In some examples, the control plane VCN 716 can be isolated from the data plane VCN 718, and the data plane mirror app tier 740 of the control plane VCN 716 can communicate with the data plane app tier 746 of the data plane VCN 718 via VNICs 742 that can be contained in the data plane mirror app tier 740 and the data plane app tier 746.

[0152] In some examples, users of the system, or customers, can make requests, for example create, read, update, or delete (CRUD) operations, through public Internet 754 that can communicate the requests to the metadata management service 752. The metadata management service 752 can communicate the request to the control plane VCN 716 through the Internet gateway 734. The request can be received by the LB subnet(s) 722 contained in the control plane DMZ tier 720. The LB subnet(s) 722 may determine that the request is valid, and in response to this determination, the LB subnet(s) 722 can transmit the request to app subnet(s) 726 contained in the control plane app tier 724. If the request is validated and requires a call to public Internet 754, the call to public Internet 754 may be transmitted to the NAT gateway 738 that can make the call to public Internet

754. Memory that may be desired to be stored by the request can be stored in the DB subnet(s) 730.

[0153] In some examples, the data plane mirror app tier 740 can facilitate direct communication between the control plane VCN 716 and the data plane VCN 718. For example, changes, updates, or other suitable modifications to configuration may be desired to be applied to the resources contained in the data plane VCN 718. Via a VNIC 742, the control plane VCN 716 can directly communicate with, and can thereby execute the changes, updates, or other suitable modifications to configuration to, resources contained in the data plane VCN 718.

[0154] In some embodiments, the control plane VCN 716 and the data plane VCN 718 can be contained in the service tenancy 719. In this case, the user, or the customer, of the system may not own or operate either the control plane VCN 716 or the data plane VCN 718. Instead, the IaaS provider may own or operate the control plane VCN 716 and the data plane VCN 718, both of which may be contained in the service tenancy 719. This embodiment can enable isolation of networks that may prevent users or customers from interacting with other users', or other customers', resources. Also, this embodiment may allow users or customers of the system to store databases privately without needing to rely on public Internet 754, which may not have a desired level of threat prevention, for storage.

[0155] In other embodiments, the LB subnet(s) 722 contained in the control plane VCN 716 can be configured to receive a signal from the service gateway 736. In this embodiment, the control plane VCN 716 and the data plane VCN 718 may be configured to be called by a customer of the IaaS provider without calling public Internet 754. Customers of the IaaS provider may desire this embodiment since database(s) that the customers use may be controlled by the IaaS provider and may be stored on the service tenancy 719, which may be isolated from public Internet 754.

[0156] FIG. 8 is a block diagram 800 illustrating another example pattern of an IaaS architecture, according to at least one embodiment. Service operators 802 (e.g. service operators 702 of FIG. 7) can be communicatively coupled to a secure host tenancy 804 (e.g. the secure host tenancy 704 of FIG. 7) that can include a virtual cloud network (VCN) 806 (e.g. the VCN 706 of FIG. 7) and a secure host subnet 808 (e.g. the secure

host subnet 708 of FIG. 7). The VCN 806 can include a local peering gateway (LPG) 810 (e.g. the LPG 710 of FIG. 7) that can be communicatively coupled to a secure shell (SSH) VCN 812 (e.g. the SSH VCN 712 of FIG. 7) via an LPG 710 contained in the SSH VCN 812. The SSH VCN 812 can include an SSH subnet 814 (e.g. the SSH subnet 714 of FIG. 7), and the SSH VCN 812 can be communicatively coupled to a control plane VCN 816 (e.g. the control plane VCN 716 of FIG. 7) via an LPG 810 contained in the control plane VCN 816. The control plane VCN 816 can be contained in a service tenancy 819 (e.g. the service tenancy 719 of FIG. 7), and the data plane VCN 818 (e.g. the data plane VCN 718 of FIG. 7) can be contained in a customer tenancy 821 that may be owned or operated by users, or customers, of the system.

[0157] The control plane VCN 816 can include a control plane DMZ tier 820 (e.g. the control plane DMZ tier 720 of FIG. 7) that can include LB subnet(s) 822 (e.g. LB subnet(s) 722 of FIG. 7), a control plane app tier 824 (e.g. the control plane app tier 724 of FIG. 7) that can include app subnet(s) 826 (e.g. app subnet(s) 726 of FIG. 7), a control plane data tier 828 (e.g. the control plane data tier 728 of FIG. 7) that can include database (DB) subnet(s) 830 (e.g. similar to DB subnet(s) 730 of FIG. 7). The LB subnet(s) 822 contained in the control plane DMZ tier 820 can be communicatively coupled to the app subnet(s) 826 contained in the control plane app tier 824 and an Internet gateway 834 (e.g. the Internet gateway 734 of FIG. 7) that can be contained in the control plane VCN 816, and the app subnet(s) 826 can be communicatively coupled to the DB subnet(s) 830 contained in the control plane data tier 828 and a service gateway 836 (e.g. the service gateway of FIG. 7) and a network address translation (NAT) gateway 838 (e.g. the NAT gateway 738 of FIG. 7). The control plane VCN 816 can include the service gateway 836 and the NAT gateway 838.

[0158] The control plane VCN 816 can include a data plane mirror app tier 840 (e.g. the data plane mirror app tier 740 of FIG. 7) that can include app subnet(s) 826. The app subnet(s) 826 contained in the data plane mirror app tier 840 can include a virtual network interface controller (VNIC) 842 (e.g. the VNIC of 742) that can execute a compute instance 844 (e.g. similar to the compute instance 744 of FIG. 7). The compute instance 844 can facilitate communication between the app subnet(s) 826 of the data plane mirror app tier 840 and the app subnet(s) 826 that can be contained in a data plane

app tier 846 (e.g. the data plane app tier 746 of FIG. 7) via the VNIC 842 contained in the data plane mirror app tier 840 and the VNIC 842 contained in the data plane app tier 846.

[0159] The Internet gateway 834 contained in the control plane VCN 816 can be communicatively coupled to a metadata management service 852 (e.g. the metadata management service 752 of FIG. 7) that can be communicatively coupled to public Internet 854 (e.g. public Internet 754 of FIG. 7). Public Internet 854 can be communicatively coupled to the NAT gateway 838 contained in the control plane VCN 816. The service gateway 836 contained in the control plane VCN 816 can be communicatively couple to cloud services 856 (e.g. cloud services 756 of FIG. 7).

[0160] In some examples, the data plane VCN 818 can be contained in the customer tenancy 821. In this case, the IaaS provider may provide the control plane VCN 816 for each customer, and the IaaS provider may, for each customer, set up a unique compute instance 844 that is contained in the service tenancy 819. Each compute instance 844 may allow communication between the control plane VCN 816, contained in the service tenancy 819, and the data plane VCN 818 that is contained in the customer tenancy 821. The compute instance 844 may allow resources, that are provisioned in the control plane VCN 816 that is contained in the service tenancy 819, to be deployed or otherwise used in the data plane VCN 818 that is contained in the customer tenancy 821.

[0161] In other examples, the customer of the IaaS provider may have databases that live in the customer tenancy 821. In this example, the control plane VCN 816 can include the data plane mirror app tier 840 that can include app subnet(s) 826. The data plane mirror app tier 840 can reside in the data plane VCN 818, but the data plane mirror app tier 840 may not live in the data plane VCN 818. That is, the data plane mirror app tier 840 may have access to the customer tenancy 821, but the data plane mirror app tier 840 may not exist in the data plane VCN 818 or be owned or operated by the customer of the IaaS provider. The data plane mirror app tier 840 may be configured to make calls to the data plane VCN 818 but may not be configured to make calls to any entity contained in the control plane VCN 816. The customer may desire to deploy or otherwise use resources in the data plane VCN 818 that are provisioned in the control plane VCN 816, and the data plane mirror app tier 840 can facilitate the desired deployment, or other usage of resources, of the customer.

[0162] In some embodiments, the customer of the IaaS provider can apply filters to the data plane VCN 818. In this embodiment, the customer can determine what the data plane VCN 818 can access, and the customer may restrict access to public Internet 854 from the data plane VCN 818. The IaaS provider may not be able to apply filters or otherwise control access of the data plane VCN 818 to any outside networks or databases. Applying filters and controls by the customer onto the data plane VCN 818, contained in the customer tenancy 821, can help isolate the data plane VCN 818 from other customers and from public Internet 854.

[0163] In some embodiments, cloud services 856 can be called by the service gateway 836 to access services that may not exist on public Internet 854, on the control plane VCN 816, or on the data plane VCN 818. The connection between cloud services 856 and the control plane VCN 816 or the data plane VCN 818 may not be live or continuous. Cloud services 856 may exist on a different network owned or operated by the IaaS provider. Cloud services 856 may be configured to receive calls from the service gateway 836 and may be configured to not receive calls from public Internet 854. Some cloud services 856 may be isolated from other cloud services 856, and the control plane VCN 816 may be isolated from cloud services 856 that may not be in the same region as the control plane VCN 816. For example, the control plane VCN 816 may be located in "Region 1," and cloud service "Deployment 7," may be located in Region 1 and in "Region 2." If a call to Deployment 7 is made by the service gateway 836 contained in the control plane VCN 816 located in Region 1, the call may be transmitted to Deployment 7 in Region 1. In this example, the control plane VCN 816, or Deployment 7 in Region 1, may not be communicatively coupled to, or otherwise in communication with, Deployment 7 in Region 2.

[0164] FIG. 9 is a block diagram 900 illustrating another example pattern of an IaaS architecture, according to at least one embodiment. Service operators 902 (e.g. service operators 702 of FIG. 7) can be communicatively coupled to a secure host tenancy 904 (e.g. the secure host tenancy 704 of FIG. 7) that can include a virtual cloud network (VCN) 906 (e.g. the VCN 706 of FIG. 7) and a secure host subnet 908 (e.g. the secure host subnet 708 of FIG. 7). The VCN 906 can include an LPG 910 (e.g. the LPG 710 of FIG. 7) that can be communicatively coupled to an SSH VCN 912 (e.g. the SSH VCN

712 of FIG. 7) via an LPG 910 contained in the SSH VCN 912. The SSH VCN 912 can include an SSH subnet 914 (e.g. the SSH subnet 714 of FIG. 7), and the SSH VCN 912 can be communicatively coupled to a control plane VCN 916 (e.g. the control plane VCN 716 of FIG. 7) via an LPG 910 contained in the control plane VCN 916 and to a data plane VCN 918 (e.g. the data plane 718 of FIG. 7) via an LPG 910 contained in the data plane VCN 918. The control plane VCN 916 and the data plane VCN 918 can be contained in a service tenancy 919 (e.g. the service tenancy 719 of FIG. 7).

[0165] The control plane VCN 916 can include a control plane DMZ tier 920 (e.g. the control plane DMZ tier 720 of FIG. 7) that can include load balancer (LB) subnet(s) 922 (e.g. LB subnet(s) 722 of FIG. 7), a control plane app tier 924 (e.g. the control plane app tier 724 of FIG. 7) that can include app subnet(s) 926 (e.g. similar to app subnet(s) 726 of FIG. 7), a control plane data tier 928 (e.g. the control plane data tier 728 of FIG. 7) that can include DB subnet(s) 930. The LB subnet(s) 922 contained in the control plane DMZ tier 920 can be communicatively coupled to the app subnet(s) 926 contained in the control plane app tier 924 and to an Internet gateway 934 (e.g. the Internet gateway 734 of FIG. 7) that can be contained in the control plane VCN 916, and the app subnet(s) 926 can be communicatively coupled to the DB subnet(s) 930 contained in the control plane data tier 928 and to a service gateway 936 (e.g. the service gateway of FIG. 7) and a network address translation (NAT) gateway 938 (e.g. the NAT gateway 738 of FIG. 7). The control plane VCN 916 can include the service gateway 936 and the NAT gateway 938.

[0166] The data plane VCN 918 can include a data plane app tier 946 (e.g. the data plane app tier 746 of FIG. 7), a data plane DMZ tier 948 (e.g. the data plane DMZ tier 748 of FIG. 7), and a data plane data tier 950 (e.g. the data plane data tier 750 of FIG. 7). The data plane DMZ tier 948 can include LB subnet(s) 922 that can be communicatively coupled to trusted app subnet(s) 960 and untrusted app subnet(s) 962 of the data plane app tier 946 and the Internet gateway 934 contained in the data plane VCN 918. The trusted app subnet(s) 960 can be communicatively coupled to the service gateway 936 contained in the data plane VCN 918, the NAT gateway 938 contained in the data plane VCN 918, and DB subnet(s) 930 contained in the data plane data tier 950. The untrusted app subnet(s) 962 can be communicatively coupled to the service gateway 936 contained in the data plane VCN 918 and DB subnet(s) 930 contained in the data plane data tier

950. The data plane data tier 950 can include DB subnet(s) 930 that can be communicatively coupled to the service gateway 936 contained in the data plane VCN 918.

[0167] The untrusted app subnet(s) 962 can include one or more primary VNICs 964(1)-(N) that can be communicatively coupled to tenant virtual machines (VMs) 966(1)-(N). Each tenant VM 966(1)-(N) can be communicatively coupled to a respective app subnet 967(1)-(N) that can be contained in respective container egress VCNs 968(1)-(N) that can be contained in respective customer tenancies 970(1)-(N). Respective secondary VNICs 972(1)-(N) can facilitate communication between the untrusted app subnet(s) 962 contained in the data plane VCN 918 and the app subnet contained in the container egress VCNs 968(1)-(N). Each container egress VCNs 968(1)-(N) can include a NAT gateway 938 that can be communicatively coupled to public Internet 954 (e.g. public Internet 754 of FIG. 7).

[0168] The Internet gateway 934 contained in the control plane VCN 916 and contained in the data plane VCN 918 can be communicatively coupled to a metadata management service 952 (e.g. the metadata management system 752 of FIG. 7) that can be communicatively coupled to public Internet 954. Public Internet 954 can be communicatively coupled to the NAT gateway 938 contained in the control plane VCN 916 and contained in the data plane VCN 918. The service gateway 936 contained in the control plane VCN 916 and contained in the data plane VCN 918 can be communicatively couple to cloud services 956.

[0169] In some embodiments, the data plane VCN 918 can be integrated with customer tenancies 970. This integration can be useful or desirable for customers of the IaaS provider in some cases such as a case that may desire support when executing code. The customer may provide code to run that may be destructive, may communicate with other customer resources, or may otherwise cause undesirable effects. In response to this, the IaaS provider may determine whether to run code given to the IaaS provider by the customer.

[0170] In some examples, the customer of the IaaS provider may grant temporary network access to the IaaS provider and request a function to be attached to the data plane tier app 946. Code to run the function may be executed in the VMs 966(1)-(N), and the

code may not be configured to run anywhere else on the data plane VCN 918. Each VM 966(1)-(N) may be connected to one customer tenancy 970. Respective containers 971(1)-(N) contained in the VMs 966(1)-(N) may be configured to run the code. In this case, there can be a dual isolation (e.g., the containers 971(1)-(N) running code, where the containers 971(1)-(N) may be contained in at least the VM 966(1)-(N) that are contained in the untrusted app subnet(s) 962), which may help prevent incorrect or otherwise undesirable code from damaging the network of the IaaS provider or from damaging a network of a different customer. The containers 971(1)-(N) may be communicatively coupled to the customer tenancy 970 and may be configured to transmit or receive data from the customer tenancy 970. The containers 971(1)-(N) may not be configured to transmit or receive data from any other entity in the data plane VCN 918. Upon completion of running the code, the IaaS provider may kill or otherwise dispose of the containers 971(1)-(N).

[0171] In some embodiments, the trusted app subnet(s) 960 may run code that may be owned or operated by the IaaS provider. In this embodiment, the trusted app subnet(s) 960 may be communicatively coupled to the DB subnet(s) 930 and be configured to execute CRUD operations in the DB subnet(s) 930. The untrusted app subnet(s) 962 may be communicatively coupled to the DB subnet(s) 930, but in this embodiment, the untrusted app subnet(s) may be configured to execute read operations in the DB subnet(s) 930. The containers 971(1)-(N) that can be contained in the VM 966(1)-(N) of each customer and that may run code from the customer may not be communicatively coupled with the DB subnet(s) 930.

[0172] In other embodiments, the control plane VCN 916 and the data plane VCN 918 may not be directly communicatively coupled. In this embodiment, there may be no direct communication between the control plane VCN 916 and the data plane VCN 918. However, communication can occur indirectly through at least one method. An LPG 910 may be established by the IaaS provider that can facilitate communication between the control plane VCN 916 and the data plane VCN 918. In another example, the control plane VCN 916 or the data plane VCN 918 can make a call to cloud services 956 via the service gateway 936. For example, a call to cloud services 956 from the control plane

VCN 916 can include a request for a service that can communicate with the data plane VCN 918.

[0173] FIG. 10 is a block diagram 1000 illustrating another example pattern of an IaaS architecture, according to at least one embodiment. Service operators 1002 (e.g. service operators 702 of FIG. 7) can be communicatively coupled to a secure host tenancy 1004 (e.g. the secure host tenancy 704 of FIG. 7) that can include a virtual cloud network (VCN) 1006 (e.g. the VCN 706 of FIG. 7) and a secure host subnet 1008 (e.g. the secure host subnet 708 of FIG. 7). The VCN 1006 can include an LPG 1010 (e.g. the LPG 710 of FIG. 7) that can be communicatively coupled to an SSH VCN 1012 (e.g. the SSH VCN 712 of FIG. 7) via an LPG 1010 contained in the SSH VCN 1012. The SSH VCN 1012 can include an SSH subnet 1014 (e.g. the SSH subnet 714 of FIG. 7), and the SSH VCN 1012 can be communicatively coupled to a control plane VCN 1016 (e.g. the control plane VCN 716 of FIG. 7) via an LPG 1010 contained in the control plane VCN 1016 and to a data plane VCN 1018 (e.g. the data plane 718 of FIG. 7) via an LPG 1010 contained in the data plane VCN 1018. The control plane VCN 1016 and the data plane VCN 1018 can be contained in a service tenancy 1019 (e.g. the service tenancy 719 of FIG. 7).

[0174] The control plane VCN 1016 can include a control plane DMZ tier 1020 (e.g. the control plane DMZ tier 720 of FIG. 7) that can include LB subnet(s) 1022 (e.g. LB subnet(s) 722 of FIG. 7), a control plane app tier 1024 (e.g. the control plane app tier 724 of FIG. 7) that can include app subnet(s) 1026 (e.g. app subnet(s) 726 of FIG. 7), a control plane data tier 1028 (e.g. the control plane data tier 728 of FIG. 7) that can include DB subnet(s) 1030 (e.g. DB subnet(s) 930 of FIG. 9). The LB subnet(s) 1022 contained in the control plane DMZ tier 1020 can be communicatively coupled to the app subnet(s) 1026 contained in the control plane app tier 1024 and to an Internet gateway 1034 (e.g. the Internet gateway 734 of FIG. 7) that can be contained in the control plane VCN 1016, and the app subnet(s) 1026 can be communicatively coupled to the DB subnet(s) 1030 contained in the control plane data tier 1028 and to a service gateway 1036 (e.g. the service gateway of FIG. 7) and a network address translation (NAT) gateway 1038 (e.g. the NAT gateway 738 of FIG. 7). The control plane VCN 1016 can include the service gateway 1036 and the NAT gateway 1038.

[0175] The data plane VCN 1018 can include a data plane app tier 1046 (e.g. the data plane app tier 746 of FIG. 7), a data plane DMZ tier 1048 (e.g. the data plane DMZ tier 748 of FIG. 7), and a data plane data tier 1050 (e.g. the data plane data tier 750 of FIG. 7). The data plane DMZ tier 1048 can include LB subnet(s) 1022 that can be communicatively coupled to trusted app subnet(s) 1060 (e.g. trusted app subnet(s) 960 of FIG. 9) and untrusted app subnet(s) 1062 (e.g. untrusted app subnet(s) 962 of FIG. 9) of the data plane app tier 1046 and the Internet gateway 1034 contained in the data plane VCN 1018. The trusted app subnet(s) 1060 can be communicatively coupled to the service gateway 1036 contained in the data plane VCN 1018, the NAT gateway 1038 contained in the data plane VCN 1018, and DB subnet(s) 1030 contained in the data plane data tier 1050. The untrusted app subnet(s) 1062 can be communicatively coupled to the service gateway 1036 contained in the data plane VCN 1018 and DB subnet(s) 1030 contained in the data plane data tier 1050. The data plane data tier 1050 can include DB subnet(s) 1030 that can be communicatively coupled to the service gateway 1036 contained in the data plane VCN 1018.

[0176] The untrusted app subnet(s) 1062 can include primary VNICs 1064(1)-(N) that can be communicatively coupled to tenant virtual machines (VMs) 1066(1)-(N) residing within the untrusted app subnet(s) 1062. Each tenant VM 1066(1)-(N) can run code in a respective container 1067(1)-(N), and be communicatively coupled to an app subnet 1026 that can be contained in a data plane app tier 1046 that can be contained in a container egress VCN 1068. Respective secondary VNICs 1072(1)-(N) can facilitate communication between the untrusted app subnet(s) 1062 contained in the data plane VCN 1018 and the app subnet contained in the container egress VCN 1068. The container egress VCN can include a NAT gateway 1038 that can be communicatively coupled to public Internet 1054 (e.g. public Internet 754 of FIG. 7).

[0177] The Internet gateway 1034 contained in the control plane VCN 1016 and contained in the data plane VCN 1018 can be communicatively coupled to a metadata management service 1052 (e.g. the metadata management system 752 of FIG. 7) that can be communicatively coupled to public Internet 1054. Public Internet 1054 can be communicatively coupled to the NAT gateway 1038 contained in the control plane VCN 1016 and contained in the data plane VCN 1018. The service gateway 1036 contained in

the control plane VCN 1016 and contained in the data plane VCN 1018 can be communicatively couple to cloud services 1056.

[0178] In some examples, the pattern illustrated by the architecture of block diagram 1000 of FIG. 10 may be considered an exception to the pattern illustrated by the architecture of block diagram 900 of FIG. 9 and may be desirable for a customer of the IaaS provider if the IaaS provider cannot directly communicate with the customer (e.g., a disconnected region). The respective containers 1067(1)-(N) that are contained in the VMs 1066(1)-(N) for each customer can be accessed in real-time by the customer. The containers 1067(1)-(N) may be configured to make calls to respective secondary VNICs 1072(1)-(N) contained in app subnet(s) 1026 of the data plane app tier 1046 that can be contained in the container egress VCN 1068. The secondary VNICs 1072(1)-(N) can transmit the calls to the NAT gateway 1038 that may transmit the calls to public Internet 1054. In this example, the containers 1067(1)-(N) that can be accessed in real-time by the customer can be isolated from the control plane VCN 1016 and can be isolated from other entities contained in the data plane VCN 1018. The containers 1067(1)-(N) may also be isolated from resources from other customers.

[0179] In other examples, the customer can use the containers 1067(1)-(N) to call cloud services 1056. In this example, the customer may run code in the containers 1067(1)-(N) that requests a service from cloud services 1056. The containers 1067(1)-(N) can transmit this request to the secondary VNICs 1072(1)-(N) that can transmit the request to the NAT gateway that can transmit the request to public Internet 1054. Public Internet 1054 can transmit the request to LB subnet(s) 1022 contained in the control plane VCN 1016 via the Internet gateway 1034. In response to determining the request is valid, the LB subnet(s) can transmit the request to app subnet(s) 1026 that can transmit the request to cloud services 1056 via the service gateway 1036.

[0180] It should be appreciated that IaaS architectures 700, 800, 900, 1000 depicted in the figures may have other components than those depicted. Further, the embodiments shown in the figures are only some examples of a cloud infrastructure system that may incorporate an embodiment of the disclosure. In some other embodiments, the IaaS systems may have more or fewer components than shown in the figures, may combine

two or more components, or may have a different configuration or arrangement of components.

[0181] In certain embodiments, the IaaS systems described herein 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 an IaaS system is the Oracle Cloud Infrastructure (OCI) provided by the present assignee.

[0182] FIG. 11 illustrates an example computer system 1100, in which various embodiments may be implemented. The system 1100 may be used to implement any of the computer systems described above. As shown in the figure, computer system 1100 includes a processing unit 1104 that communicates with a number of peripheral subsystems via a bus subsystem 1102. These peripheral subsystems may include a processing acceleration unit 1106, an I/O subsystem 1108, a storage subsystem 1118 and a communications subsystem 1124. Storage subsystem 1118 includes tangible computer-readable storage media 1122 and a system memory 1110.

[0183] Bus subsystem 1102 provides a mechanism for letting the various components and subsystems of computer system 1100 communicate with each other as intended. Although bus subsystem 1102 is shown schematically as a single bus, alternative embodiments of the bus subsystem may utilize multiple buses. Bus subsystem 1102 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. For example, such architectures may include an Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus, which can be implemented as a Mezzanine bus manufactured to the IEEE P1386.1 standard.

[0184] Processing unit 1104, which can be implemented as one or more integrated circuits (e.g., a conventional microprocessor or microcontroller), controls the operation of computer system 1100. One or more processors may be included in processing unit 1104. These processors may include single core or multicore processors. In certain embodiments, processing unit 1104 may be implemented as one or more independent processing units 1132 and/or 1134 with single or multicore processors included in each processing unit. In other

embodiments, processing unit 1104 may also be implemented as a quad-core processing unit formed by integrating two dual-core processors into a single chip.

[0185] In various embodiments, processing unit 1104 can execute a variety of programs in response to program code and can maintain multiple concurrently executing programs or processes. At any given time, some or all of the program code to be executed can be resident in processor(s) 1104 and/or in storage subsystem 1118. Through suitable programming, processor(s) 1104 can provide various functionalities described above. Computer system 1100 may additionally include a processing acceleration unit 1106, which can include a digital signal processor (DSP), a special-purpose processor, and/or the like.

[0186] I/O subsystem 1108 may include user interface input devices and user interface output devices. User interface input devices may include a keyboard, pointing devices such as a mouse or trackball, a touchpad or touch screen incorporated into a display, a scroll wheel, a click wheel, a dial, a button, a switch, a keypad, audio input devices with voice command recognition systems, microphones, and other types of input devices. User interface input devices may include, for example, motion sensing and/or gesture recognition devices such as the Microsoft Kinect® motion sensor that enables users to control and interact with an input device, such as the Microsoft Xbox® 360 game controller, through a natural user interface using gestures and spoken commands. User interface input devices may also include eye gesture recognition devices such as the Google Glass® blink detector that detects eye activity (e.g., 'blinking' while taking pictures and/or making a menu selection) from users and transforms the eye gestures as input into an input device (e.g., Google Glass®). Additionally, user interface input devices may include voice recognition sensing devices that enable users to interact with voice recognition systems (e.g., Siri® navigator), through voice commands.

[0187] User interface input devices may also include, without limitation, three dimensional (3D) mice, joysticks or pointing sticks, gamepads and graphic tablets, and audio/visual devices such as speakers, digital cameras, digital camcorders, portable media players, webcams, image scanners, fingerprint scanners, barcode reader 3D scanners, 3D printers, laser rangefinders, and eye gaze tracking devices. Additionally, user interface input devices may include, for example, medical imaging input devices such as computed tomography, magnetic resonance imaging, position emission tomography, medical ultrasonography devices. User interface input devices may also include, for example, audio input devices such as MIDI keyboards, digital musical instruments and the like.

[0188] User interface output devices may include a display subsystem, indicator lights, or non-visual displays such as audio output devices, etc. The display subsystem may be a cathode ray tube (CRT), a flat-panel device, such as that using a liquid crystal display (LCD) or plasma display, a projection device, a touch screen, and the like. In general, use of the term "output device" is intended to include all possible types of devices and mechanisms for outputting information from computer system 1100 to a user or other computer. For example, user interface output devices may include, without limitation, a variety of display devices that visually convey text, graphics and audio/video information such as monitors, printers, speakers, headphones, automotive navigation systems, plotters, voice output devices, and modems.

[0189] Computer system 1100 may comprise a storage subsystem 1118 that provides a tangible non-transitory computer-readable storage medium for storing software and data constructs that provide the functionality of the embodiments described in this disclosure. The software can include programs, code, instructions, scripts, etc., that when executed by one or more cores or processors of processing unit 1104 provide the functionality described above. Storage subsystem 1118 may also provide a repository for storing data used in accordance with the present disclosure.

[0190] As depicted in the example in FIG. 11, storage subsystem 1118 can include various components including a system memory 1110, computer-readable storage media 1122, and a computer readable storage media reader 1120. System memory 1110 may store program instructions that are loadable and executable by processing unit 1104. System memory 1110 may also store data that is used during the execution of the instructions and/or data that is generated during the execution of the program instructions. Various different kinds of programs may be loaded into system memory 1110 including but not limited to client applications, Web browsers, mid-tier applications, relational database management systems (RDBMS), virtual machines, containers, etc.

[0191] System memory 1110 may also store an operating system 1116. Examples of operating system 1116 may include various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems, a variety of commercially-available UNIX® or UNIX-like operating systems (including without limitation the variety of GNU/Linux operating systems, the Google Chrome® OS, and the like) and/or mobile operating systems such as iOS, Windows® Phone, Android® OS, BlackBerry® OS, and Palm® OS operating

systems. In certain implementations where computer system 1100 executes one or more virtual machines, the virtual machines along with their guest operating systems (GOSs) may be loaded into system memory 1110 and executed by one or more processors or cores of processing unit 1104.

[0192] System memory 1110 can come in different configurations depending upon the type of computer system 1100. For example, system memory 1110 may be volatile memory (such as random access memory (RAM)) and/or non-volatile memory (such as read-only memory (ROM), flash memory, etc.). Different types of RAM configurations may be provided including a static random access memory (SRAM), a dynamic random access memory (DRAM), and others. In some implementations, system memory 1110 may include a basic input/output system (BIOS) containing basic routines that help to transfer information between elements within computer system 1100, such as during start-up.

[0193] Computer-readable storage media 1122 may represent remote, local, fixed, and/or removable storage devices plus storage media for temporarily and/or more permanently containing, storing, computer-readable information for use by computer system 1100 including instructions executable by processing unit 1104 of computer system 1100.

[0194] Computer-readable storage media 1122 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 and/or transmission of information. This can include tangible computer-readable storage media such as RAM, ROM, electronically erasable programmable 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 other tangible computer readable media.

[0195] By way of example, computer-readable storage media 1122 may include a hard disk drive that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive that reads from or writes to a removable, nonvolatile magnetic disk, and an optical disk drive that reads from or writes to a removable, nonvolatile optical disk such as a CD ROM, DVD, and Blu-Ray® disk, or other optical media. Computer-readable storage media 1122 may include, but is not limited to, Zip® drives, flash memory cards, universal serial bus (USB) flash drives, secure digital (SD) cards, DVD disks, digital video tape, and the like.

Computer-readable storage media 1122 may also include, solid-state drives (SSD) based on non-volatile memory such as flash-memory based SSDs, enterprise flash drives, solid state ROM, and the like, SSDs based on volatile memory such as solid state RAM, dynamic RAM, static RAM, DRAM-based SSDs, magnetoresistive RAM (MRAM) SSDs, and hybrid SSDs that use a combination of DRAM and flash memory based SSDs. The disk drives and their associated computer-readable media may provide non-volatile storage of computer-readable instructions, data structures, programs, and other data for computer system 1100.

[0196] Machine-readable instructions executable by one or more processors or cores of processing unit 1104 may be stored on a non-transitory computer-readable storage medium. A non-transitory computer-readable storage medium can include physically tangible memory or storage devices that include volatile memory storage devices and/or non-volatile storage devices. Examples of non-transitory computer-readable storage medium include magnetic storage media (e.g., disk or tapes), optical storage media (e.g., DVDs, CDs), various types of RAM, ROM, or flash memory, hard drives, floppy drives, detachable memory drives (e.g., USB drives), or other type of storage device.

[0197] Communications subsystem 1124 provides an interface to other computer systems and networks. Communications subsystem 1124 serves as an interface for receiving data from and transmitting data to other systems from computer system 1100. For example, communications subsystem 1124 may enable computer system 1100 to connect to one or more devices via the Internet. In some embodiments communications subsystem 1124 can include radio frequency (RF) transceiver components for accessing wireless voice and/or data networks (e.g., using cellular telephone technology, advanced data network technology, such as 3G, 4G or EDGE (enhanced data rates for global evolution), WiFi (IEEE 802.11 family standards, or other mobile communication technologies, or any combination thereof), global positioning system (GPS) receiver components, and/or other components. In some embodiments communications subsystem 1124 can provide wired network connectivity (e.g., Ethernet) in addition to or instead of a wireless interface.

[0198] In some embodiments, communications subsystem 1124 may also receive input communication in the form of structured and/or unstructured data feeds 1126, event streams 1128, event updates 1130, and the like on behalf of one or more users who may use computer system 1100.

[0199] By way of example, communications subsystem 1124 may be configured to receive data feeds 1126 in real-time from users of social networks and/or other communication services such as Twitter® feeds, Facebook® updates, web feeds such as Rich Site Summary (RSS) feeds, and/or real-time updates from one or more third party information sources.

[0200] Additionally, communications subsystem 1124 may also be configured to receive data in the form of continuous data streams, which may include event streams 1128 of real-time events and/or event updates 1130, that may be continuous or unbounded in nature with no explicit end. Examples of applications that generate continuous data may include, for example, sensor data applications, financial tickers, network performance measuring tools (e.g. network monitoring and traffic management applications), clickstream analysis tools, automobile traffic monitoring, and the like.

[0201] Communications subsystem 1124 may also be configured to output the structured and/or unstructured data feeds 1126, event streams 1128, event updates 1130, and the like to one or more databases that may be in communication with one or more streaming data source computers coupled to computer system 1100.

[0202] Computer system 1100 can be one of various types, including a handheld portable device (e.g., an iPhone® cellular phone, an iPad® computing tablet, a PDA), a wearable device (e.g., a Google Glass® head mounted display), a PC, a workstation, a mainframe, a kiosk, a server rack, or any other data processing system.

[0203] Due to the ever-changing nature of computers and networks, the description of computer system 1100 depicted in the figure is intended only as a specific example. Many other configurations having more or fewer components than the system depicted in the figure are possible. For example, customized hardware might also be used and/or particular elements might be implemented in hardware, firmware, software (including applets), or a combination. Further, connection to other computing devices, such as network input/output devices, may be employed. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various embodiments.

[0204] Although specific embodiments have been described, various modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the disclosure. Embodiments 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 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 disclosure is not limited to the described series of transactions and steps. Various features and aspects of the above-described embodiments may be used individually or jointly.

[0205] Further, while embodiments 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 disclosure. Embodiments may be implemented only in hardware, or only in software, or using combinations thereof. The various processes described herein can be implemented on the same processor or different processors in any combination. Accordingly, where components are described as being configured to perform certain operations, such configuration can be accomplished, e.g., by designing electronic circuits to perform the operation, by programming programmable electronic circuits (such as microprocessors) to perform the operation, or any combination thereof. Processes can communicate using a variety of techniques including but not limited to conventional techniques for inter process communication, and different pairs of processes may use different techniques, or the same pair of processes may use different techniques at different times. Embodiments may be implemented by using a computer program product, comprising computer program/instructions which, when executed by a processor, cause the processor to perform any of the methods described in the disclosure.

[0206] 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. Thus, although specific disclosure embodiments have been described, these are not intended to be limiting. Various modifications and equivalents are within the scope of the following claims.

[0207] The use of the terms "a" and "an" and "the" and similar referents in the context of describing the disclosed embodiments (especially in the context of the following claims) are to be construed to cover both the singular and the plural, unless otherwise indicated herein or clearly contradicted by context. The terms "comprising," "having," "including," and "containing" are to be construed as open-ended terms (i.e., meaning "including, but not limited to,") unless otherwise noted. The term "connected" is to be construed as partly or

wholly contained within, attached to, or joined together, even if there is something intervening. Recitation of ranges of values herein are merely intended to serve as a shorthand method of referring individually to each separate value falling within the range, unless otherwise indicated herein and each separate value is incorporated into the specification as if it were individually recited herein. All methods described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context. The use of any and all examples, or exemplary language (e.g., "such as") provided herein, is intended merely to better illuminate embodiments and does not pose a limitation on the scope of the disclosure unless otherwise claimed. No language in the specification should be construed as indicating any non-claimed element as essential to the practice of the disclosure.

[0208] Disjunctive language such as the phrase "at least one of X, Y, or Z," unless specifically stated otherwise, is intended to be understood within the context as used in general to present that an item, term, etc., may be either X, Y, or Z, or any combination thereof (e.g., X, Y, and/or Z). Thus, such disjunctive language is not generally intended to, and should not, imply that certain embodiments require at least one of X, at least one of Y, or at least one of Z to each be present.

[0209] Preferred embodiments of this disclosure are described herein, including the best mode known for carrying out the disclosure. Variations of those preferred embodiments may become apparent to those of ordinary skill in the art upon reading the foregoing description. Those of ordinary skill should be able to employ such variations as appropriate and the disclosure may be practiced otherwise than as specifically described herein. Accordingly, this disclosure includes all modifications and equivalents of the subject matter recited in the claims appended hereto as permitted by applicable law. Moreover, any combination of the above-described elements in all possible variations thereof is encompassed by the disclosure unless otherwise indicated herein.

[0210] All references, including publications, patent applications, and patents, cited herein are hereby incorporated by reference to the same extent as if each reference were individually and specifically indicated to be incorporated by reference and were set forth in its entirety herein.

[0211] In the foregoing specification, aspects of the disclosure are described with reference to specific embodiments thereof, but those skilled in the art will recognize that the disclosure is not limited thereto. Various features and aspects of the above-described disclosure may be

used individually or jointly. Further, embodiments can be utilized in any number of environments and applications beyond those described herein without departing from the broader spirit and scope of the specification. The specification and drawings are, accordingly, to be regarded as illustrative rather than restrictive.

WHAT IS CLAIMED IS:

1. A method, comprising:

generating, by a distributed computing system of a cloud service provider, a virtual cloud network in a data center of a host region;

implementing, by the distributed computing system, a virtual bootstrap environment in the virtual cloud network, the virtual bootstrap environment comprising a plurality of services;

deploying, by the distributed computing system, an instance of one of the plurality of services of the virtual bootstrap environment to a target region data center;

receiving, from the instance of the service in the target region data center, an indication that the instance was successfully deployed;

identifying, responsive to the indication, a resource associated with the instance of the service; and

updating, by the distributed computing system, a second service of the plurality of services with the resource.

2. The method of claim 1, further comprising updating, by the distributed computing system, the instance of the service deployed to the target region data center.

3. The method of claim 1, further comprising migrating, by the distributed computing system prior to identifying the resource, data resources from the virtual bootstrap environment to the target region data center, the data resources associated with the instance of the service deployed to the target region.

4. The method of any one of claims 1 to 3, wherein deploying the instance uses a virtual private network connection between the data center of the host region and the target region data center.

5. The method of any one of claims 1 to 4, wherein the resource is a domain name service record.

6. The method of any one of claims 1 to 5, wherein the indication comprises a capability published to a capabilities service of the plurality of services, the capability indicating a successful partial deployment.

7. The method of claim 6, further comprising receiving, from the instance deployed to the target region data center responsive to the update of the second service, a second capability indicating a successful full deployment.

8. A computing system comprising:

one or more processors; and

one or more memories storing computer-executable instructions that, when executed with the one or more processors, cause the computing system to at least:

generate a virtual cloud network in a data center of a host region;

implement a virtual bootstrap environment in the virtual cloud network, the virtual bootstrap environment comprising a plurality of services;

deploy an instance of one of the plurality of services of the virtual bootstrap environment to a target region data center;

receive, from the instance of the service in the target region data center, an indication that the instance was successfully deployed;

identify, responsive to the indication, a resource associated with the instance of the service; and

update a second service of the plurality of services with the resource.

9. The computing system of claim 8, wherein the one or more memories store additional instructions that, when executed with the one or more processors, cause the computing system to further update the instance of the service deployed to the target region data center.

10. The computing system of claim 8, wherein the one or more memories store additional instructions that, when executed with the one or more processors, cause the computing system to further migrate, prior to identifying the resource, data resources from the virtual bootstrap environment to the target region data center, the data resources associated with the instance of the service deployed to the target region.

11. The computing system of any one of claims 8 to 10, wherein deploying the instance uses a virtual private network connection between the data center of the host region and the target region data center.

12. The computing system of any one of claims 8 to 11, wherein the resource is a domain name service record.

13. The computing system of any one of claims 8 to 12, wherein the indication comprises a capability published to a capabilities service of the plurality of services, the capability indicating a successful partial deployment.

14. The computing system of claim 13, wherein the one or more memories store additional instructions that, when executed with the one or more processors, cause the computing system to further receive, from the instance deployed to the target region data center responsive to the update of the second service, a second capability indicating a successful full deployment.

15. A non-transitory computer-readable medium storing computer-executable instructions that, when executed with one or more processors, cause a computing system to at least:

generate a virtual cloud network in a data center of a host region;

implement a virtual bootstrap environment in the virtual cloud network, the virtual bootstrap environment comprising a plurality of services;

deploy an instance of one of the plurality of services of the virtual bootstrap environment to a target region data center;

receive, from the instance of the service in the target region data center, an indication that the instance was successfully deployed;

identify, responsive to the indication, a resource associated with the instance of the service; and

update a second service of the plurality of services with the resource.

16. The non-transitory computer-readable medium of claim 15, storing further instructions that, when executed with the one or more processors, cause the computing system to further update the instance of the service deployed to the target region data center.

17. The non-transitory computer-readable medium of claim 15, storing further instructions that, when executed with the one or more processors, cause the computing system to further migrate, prior to identifying the resource, data resources from the virtual bootstrap environment to the target region data center, the data resources associated with the instance of the service deployed to the target region.

18. The non-transitory computer-readable medium of any one of claims 15 to 17, wherein deploying the instance uses a virtual private network connection between the data center of the host region and the target region data center.

19. The non-transitory computer-readable medium of any one of claims 15 to 18, wherein the resource is a domain name service record.

20. The non-transitory computer-readable medium of any one of claims 15 to 19, wherein the indication comprises a capability published to a capabilities service of the plurality of services, the capability indicating a successful partial deployment.

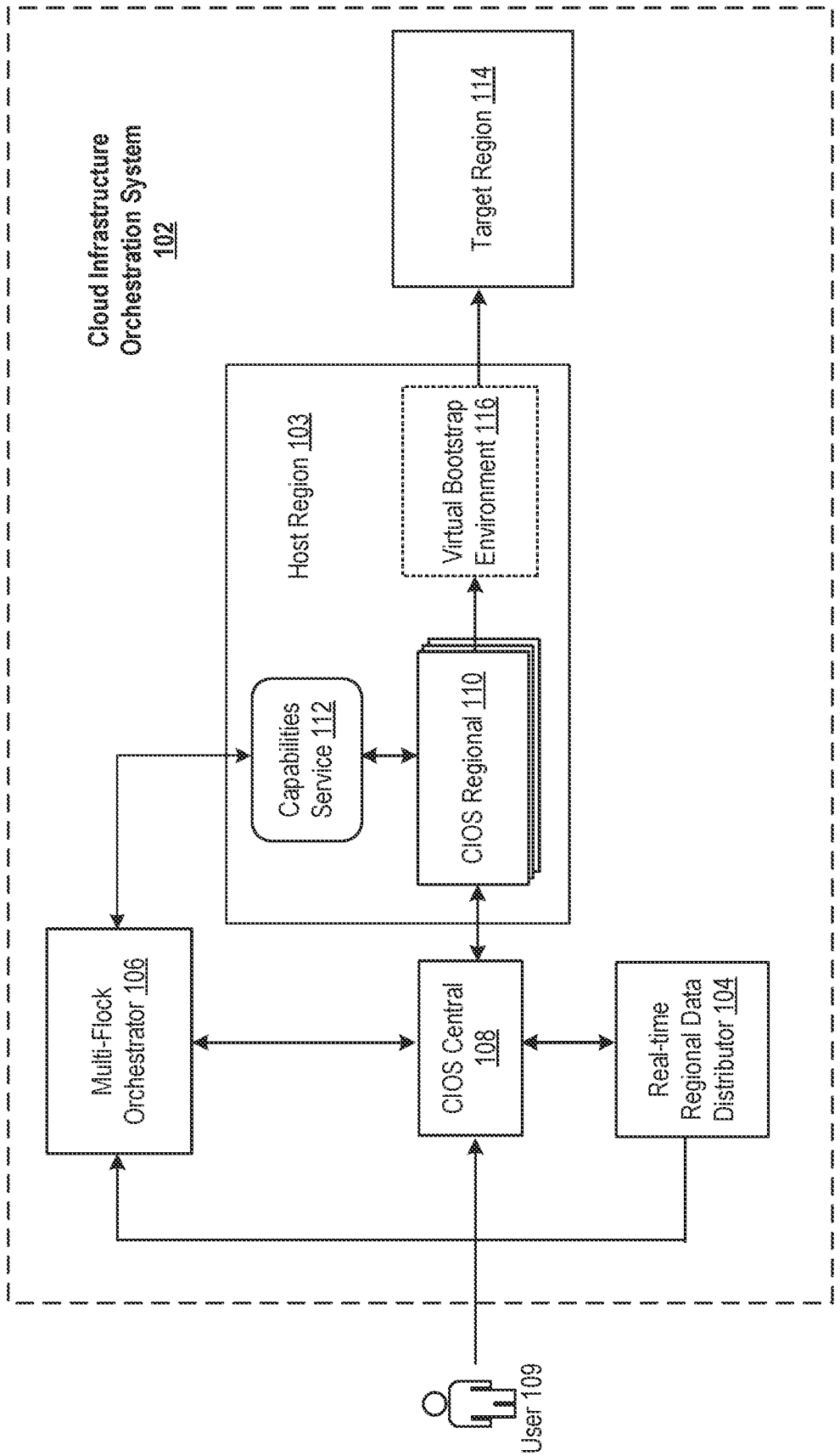

Cloud Infrastructure Orchestration System 102
Target Region 114
Host Region 103
Virtual Bootstrap Environment 116
Capabilities Service 112
CIOS Regional 110
Multi-Flock Orchestrator 106
CIOS Central 108
Real-time Regional Data Distributor 104
User 109
100

FIG. 1

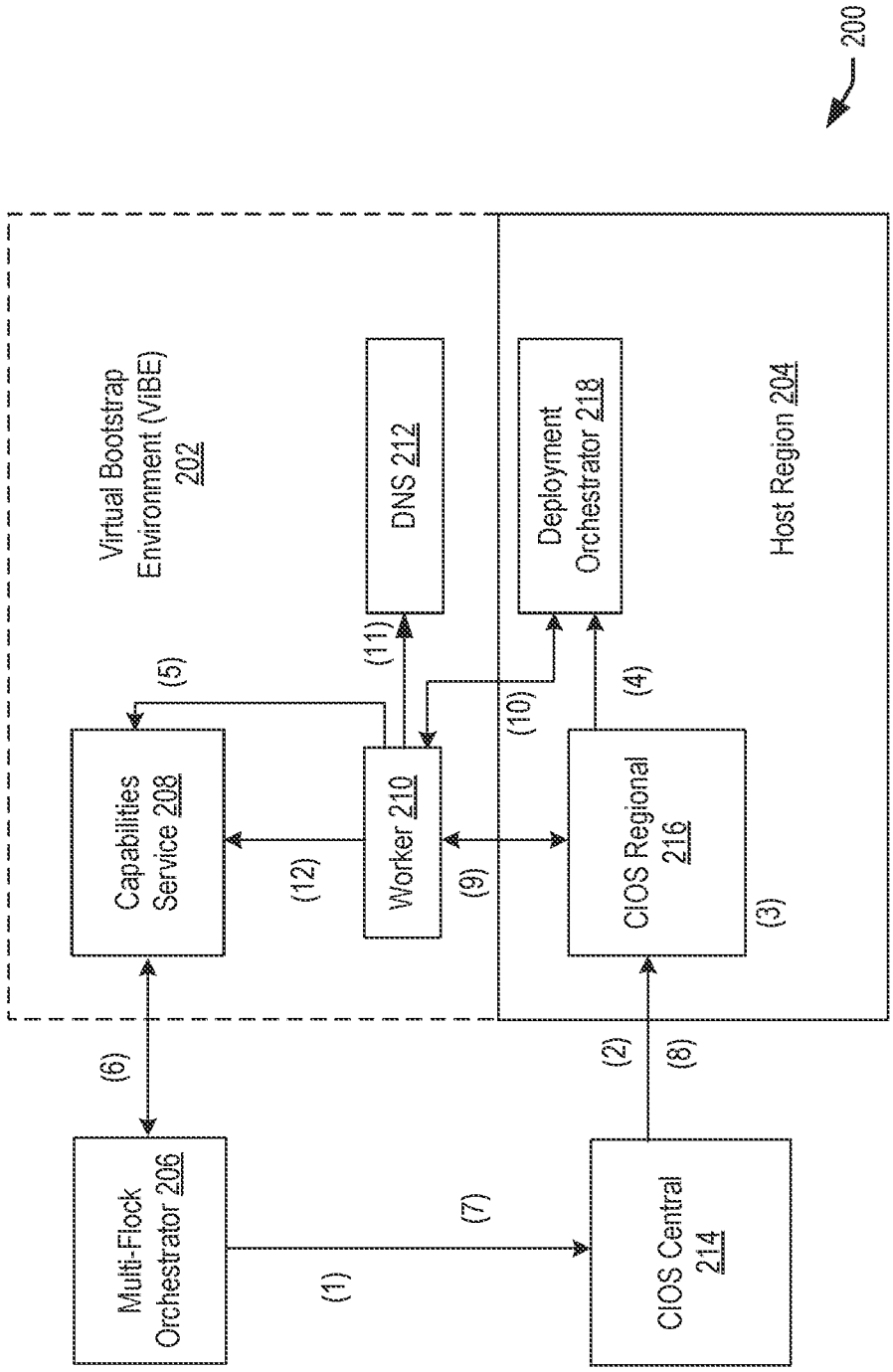

200
Virtual Bootstrap Environment (VIBE) 202
Host Region 204
Multi-Flock Orchestrator 206
Capabilities Service 208
Worker 210
DNS 212
CIOS Central 214
CIOS Regional 216
Deployment Orchestrator 218
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)

FIG. 2

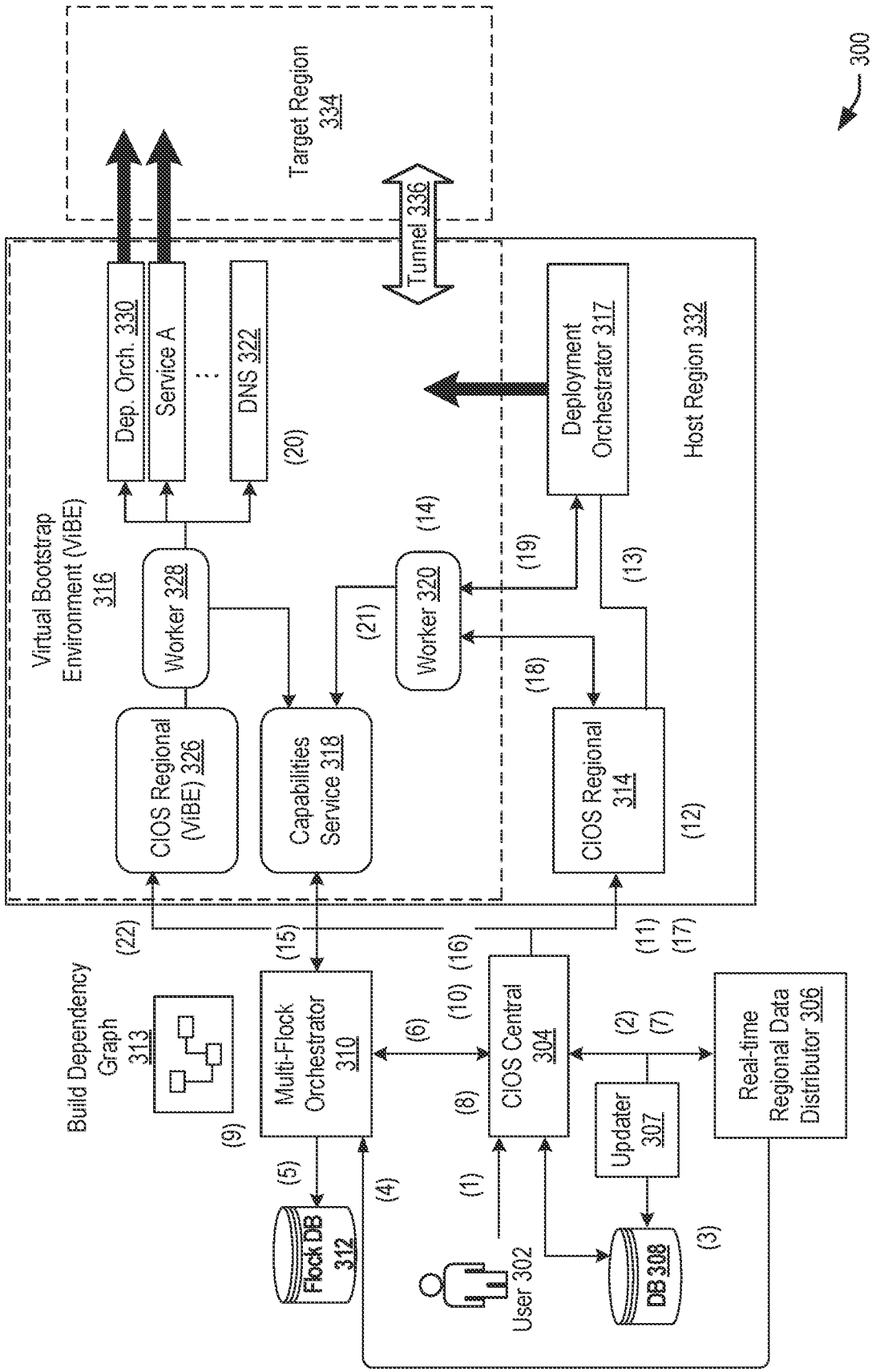

300
Target Region 334
Tunnel 336
Virtual Bootstrap Environment (ViBE) 316
Dep. Orch. 330
Service A
DNS 322
(20)
Worker 328
CIOS Regional (ViBE) 326
Capabilities Service 318
(21)
Worker 320
(14)
(19)
Deployment Orchestrator 317
Host Region 332
(18)
(13)
CIOS Regional 314
(12)
(22)
(15)
(16)
(11)
(17)
Build Dependency Graph 313
(9)
Multi-Flock Orchestrator 310
(6)
(10)
CIOS Central 304
(2)
(7)
Real-time Regional Data Distributor 306
(8)
Updater 307
(5)
(4)
(1)
Flock DB 312
User 302
DB 308
(3)

FIG. 3

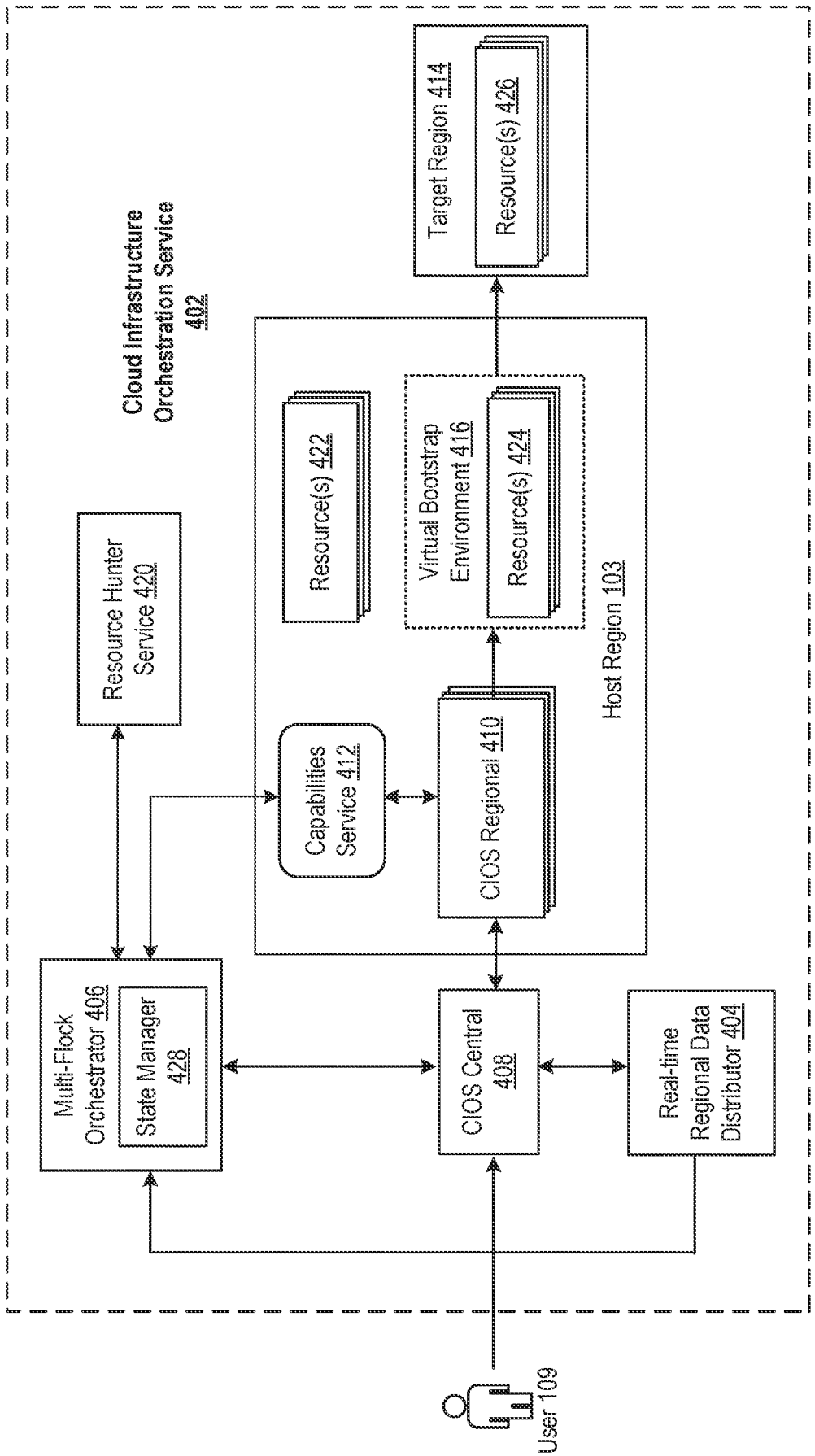

Cloud Infrastructure Orchestration Service 402
Target Region 414
Resource(s) 426
Resource Hunter Service 420
Resource(s) 422
Virtual Bootstrap Environment 416
Resource(s) 424
Host Region 103
Capabilities Service 412
CIOS Regional 410
Multi-Flock Orchestrator 406
State Manager 428
CIOS Central 408
Real-time Regional Data Distributor 404
User 109
400

FIG. 4

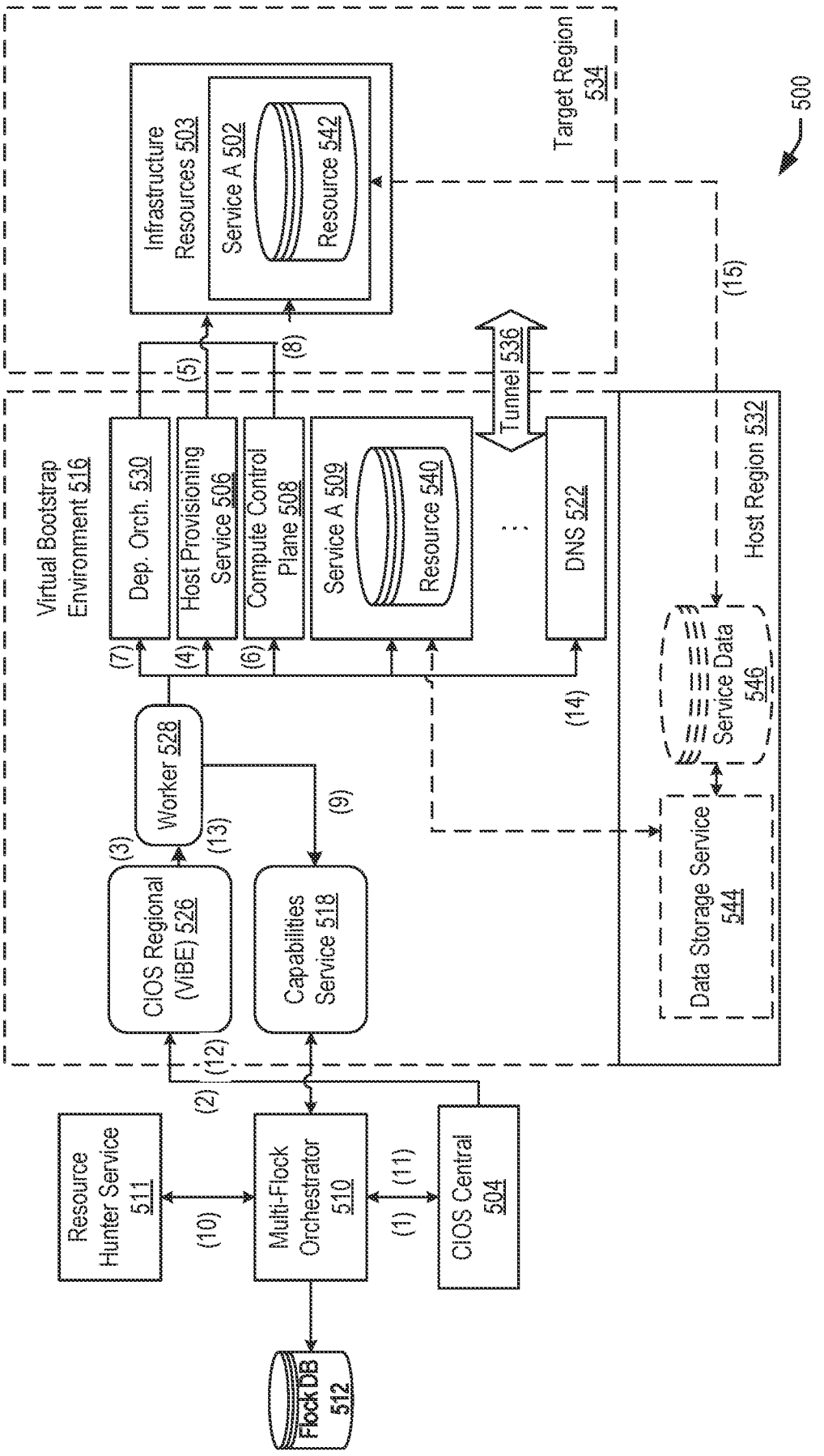

500
Target Region 534
Infrastructure Resources 503
Service A 502
Resource 542
Virtual Bootstrap Environment 516
Dep. Orch. 530
Host Provisioning Service 506
Compute Control Plane 508
Service A 509
Resource 540
Tunnel 536
DNS 522
Host Region 532
Service Data 546
Data Storage Service 544
Worker 528
CIOS Regional (VIBE) 526
Capabilities Service 518
Resource Hunter Service 511
Multi-Flock Orchestrator 510
CIOS Central 504
Flock DB 512
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)

FIG. 5

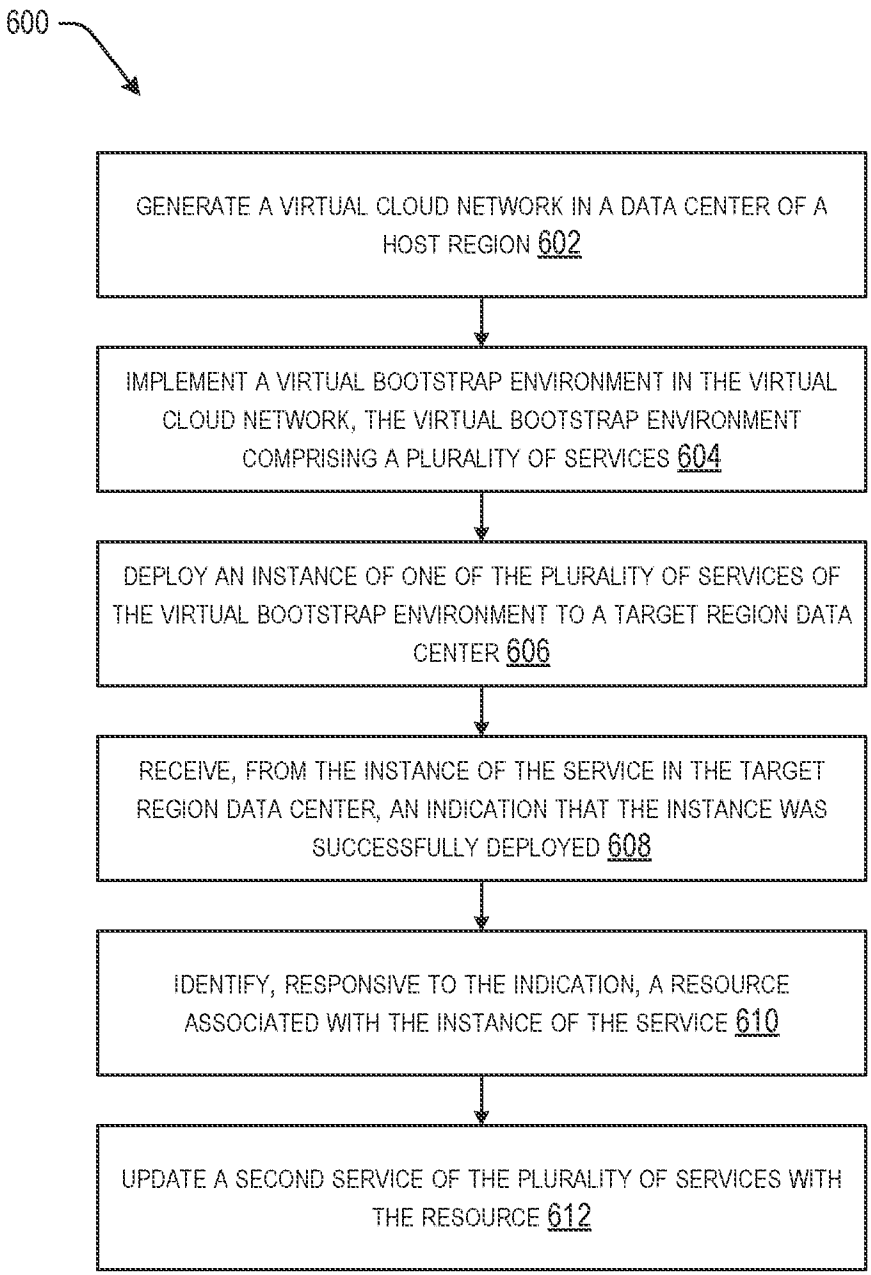

600
GENERATE A VIRTUAL CLOUD NETWORK IN A DATA CENTER OF A HOST REGION 602
IMPLEMENT A VIRTUAL BOOTSTRAP ENVIRONMENT IN THE VIRTUAL CLOUD NETWORK, THE VIRTUAL BOOTSTRAP ENVIRONMENT COMPRISING A PLURALITY OF SERVICES 604
DEPLOY AN INSTANCE OF ONE OF THE PLURALITY OF SERVICES OF THE VIRTUAL BOOTSTRAP ENVIRONMENT TO A TARGET REGION DATA CENTER 606
RECEIVE, FROM THE INSTANCE OF THE SERVICE IN THE TARGET REGION DATA CENTER, AN INDICATION THAT THE INSTANCE WAS SUCCESSFULLY DEPLOYED 608
IDENTIFY, RESPONSIVE TO THE INDICATION, A RESOURCE ASSOCIATED WITH THE INSTANCE OF THE SERVICE 610
UPDATE A SECOND SERVICE OF THE PLURALITY OF SERVICES WITH THE RESOURCE 612

FIG. 6

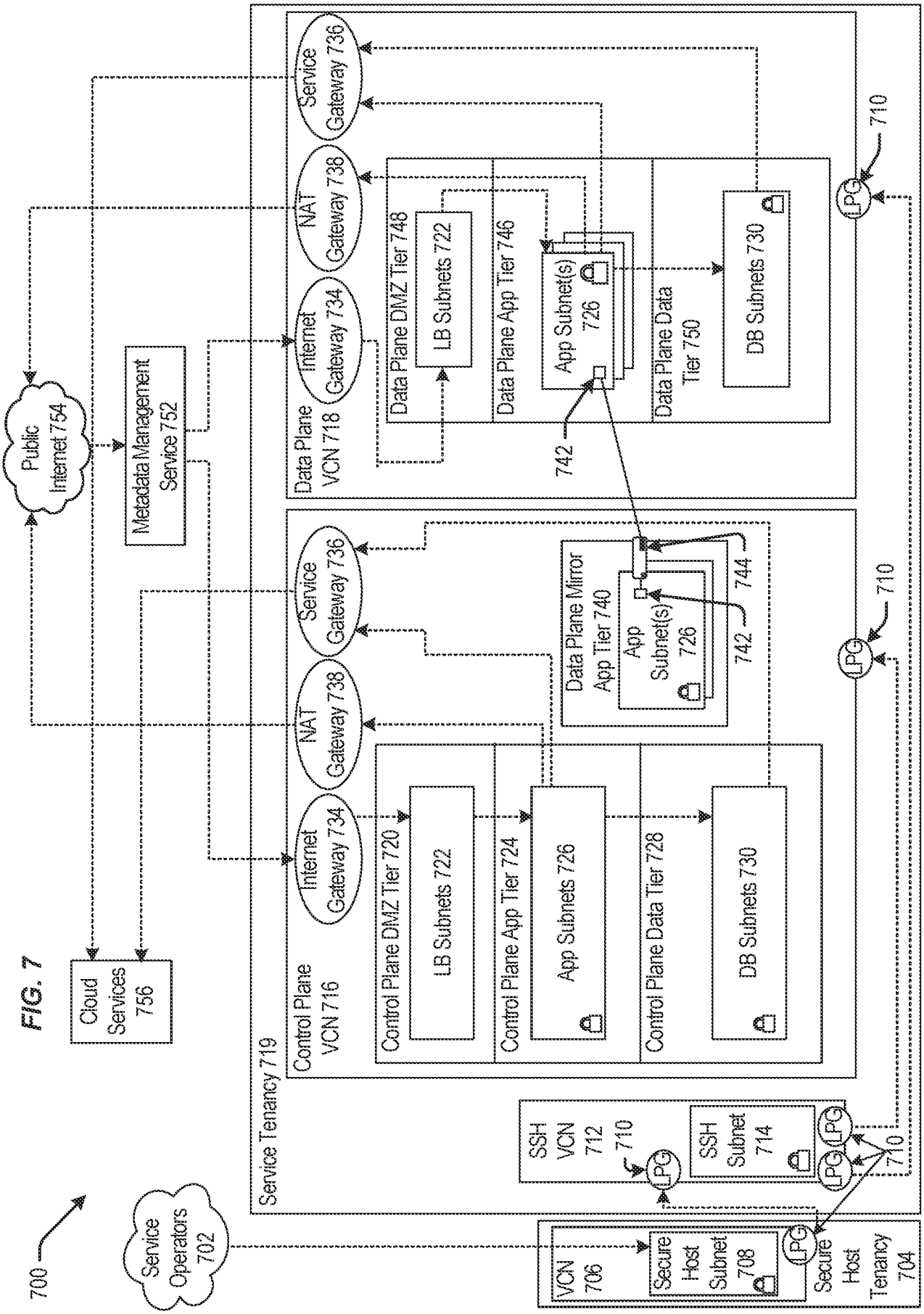
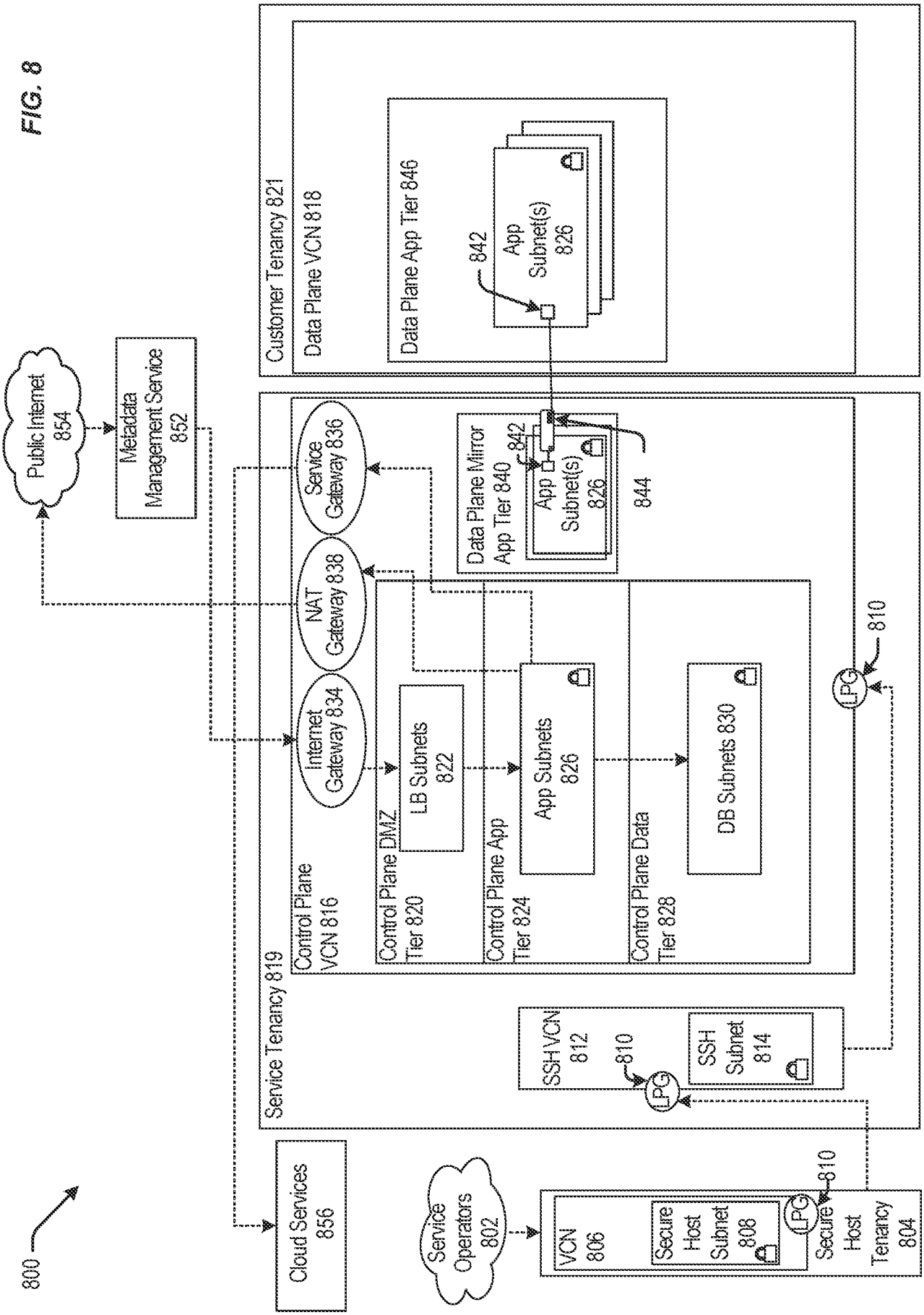

700
Service Operators 702
Public Internet 754
Metadata Management Service 752
Cloud Services 756
Service Tenancy 719
Control Plane VCN 716
Control Plane DMZ Tier 720
LB Subnets 722
Control Plane App Tier 724
App Subnets 726
Control Plane Data Tier 728
DB Subnets 730
Internet Gateway 734
NAT Gateway 738
Service Gateway 736
Data Plane Mirror App Tier 740
App Subnet(s) 726
742
744
Data Plane VCN 718
Data Plane DMZ Tier 748
LB Subnets 722
Data Plane App Tier 746
App Subnet(s) 726
742
Data Plane Data Tier 750
DB Subnets 730
Internet Gateway 734
NAT Gateway 738
Service Gateway 736
LPG
710
SSH VCN 712
SSH Subnet 714
Secure Host Tenancy 704
VCN 706
Secure Host Subnet 708

FIG. 7

FIG. 8


800
Service Operators 802
Cloud Services 856
Public Internet 854
Metadata Management Service 852
VCN 806
Secure Host Subnet 808
Secure Host Tenancy 804
SSH VCN 812
SSH Subnet 814
Service Tenancy 819
Control Plane VCN 816
Control Plane DMZ Tier 820
LB Subnets 822
Control Plane App Tier 824
App Subnets 826
Control Plane Data Tier 828
DB Subnets 830
Internet Gateway 834
NAT Gateway 838
Service Gateway 836
Data Plane Mirror App Tier 840
App Subnet(s) 826
842
844
Customer Tenancy 821
Data Plane VCN 818
Data Plane App Tier 846
App Subnet(s) 826
842
LPG
810

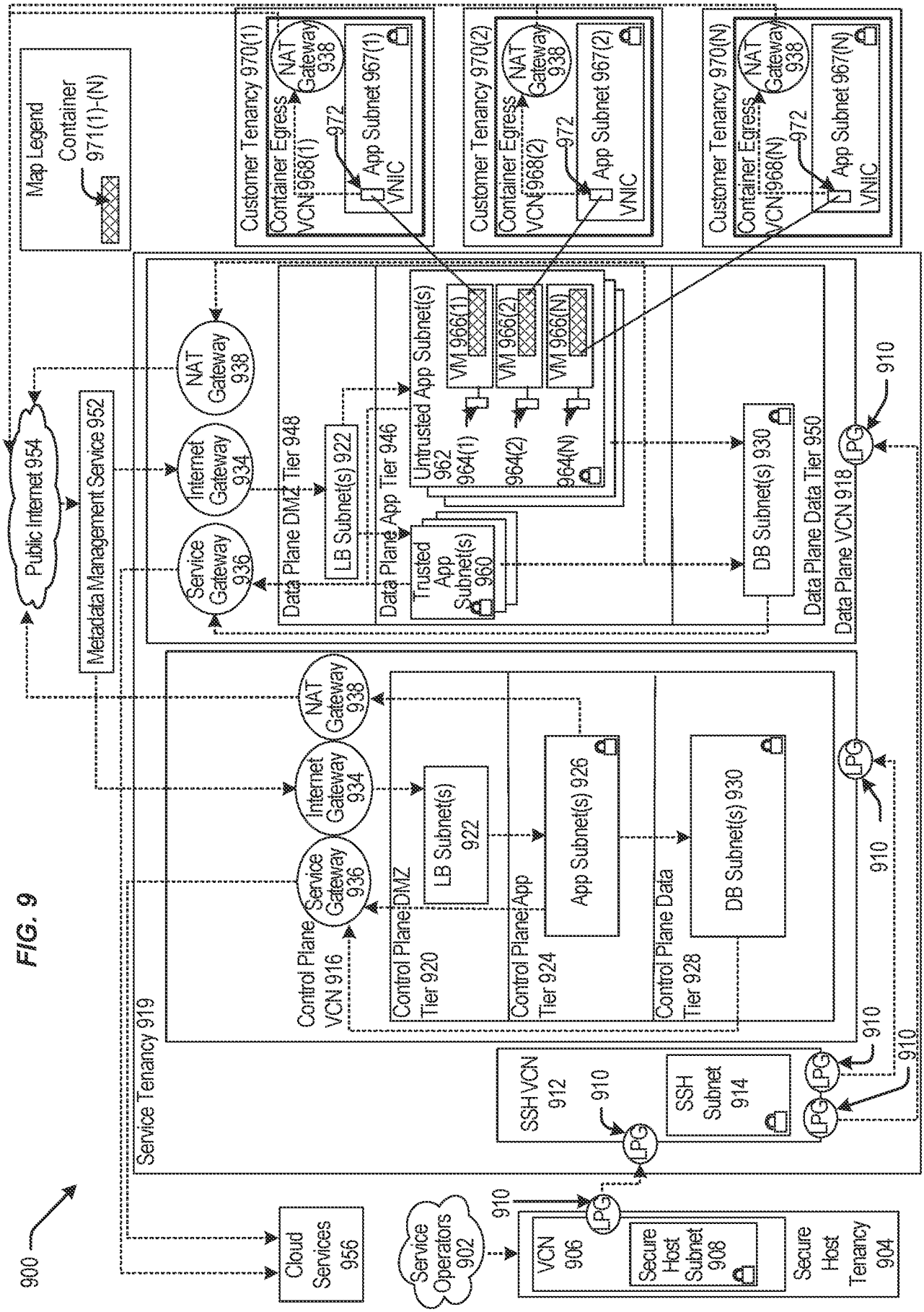

900
Map Legend
Container
971(1)-(N)
Public Internet 954
Metadata Management Service 952
Service Gateway 936
Internet Gateway 934
NAT Gateway 938
Customer Tenancy 970(1)
Container Egress VCN 968(1)
972
App Subnet 967(1) VNIC
Customer Tenancy 970(2)
Container Egress VCN 968(2)
App Subnet 967(2) VNIC
Customer Tenancy 970(N)
Container Egress VCN 968(N)
App Subnet 967(N) VNIC
Data Plane DMZ Tier 948
LB Subnet(s) 922
Data Plane App Tier 946
Untrusted App Subnet(s) 962
964(1)
964(2)
964(N)
VM 966(1)
VM 966(2)
VM 966(N)
Trusted App Subnet(s) 960
DB Subnet(s) 930
Data Plane Data Tier 950
Data Plane VCN 918
LPG
910
Service Tenancy 919
Control Plane VCN 916
Control Plane DMZ Tier 920
Control Plane App Tier 924
App Subnet(s) 926
Control Plane Data Tier 928
SSH VCN 912
SSH Subnet 914
Cloud Services 956
Service Operators 902
VCN 906
Secure Host Subnet 908
Secure Host Tenancy 904

FIG. 9

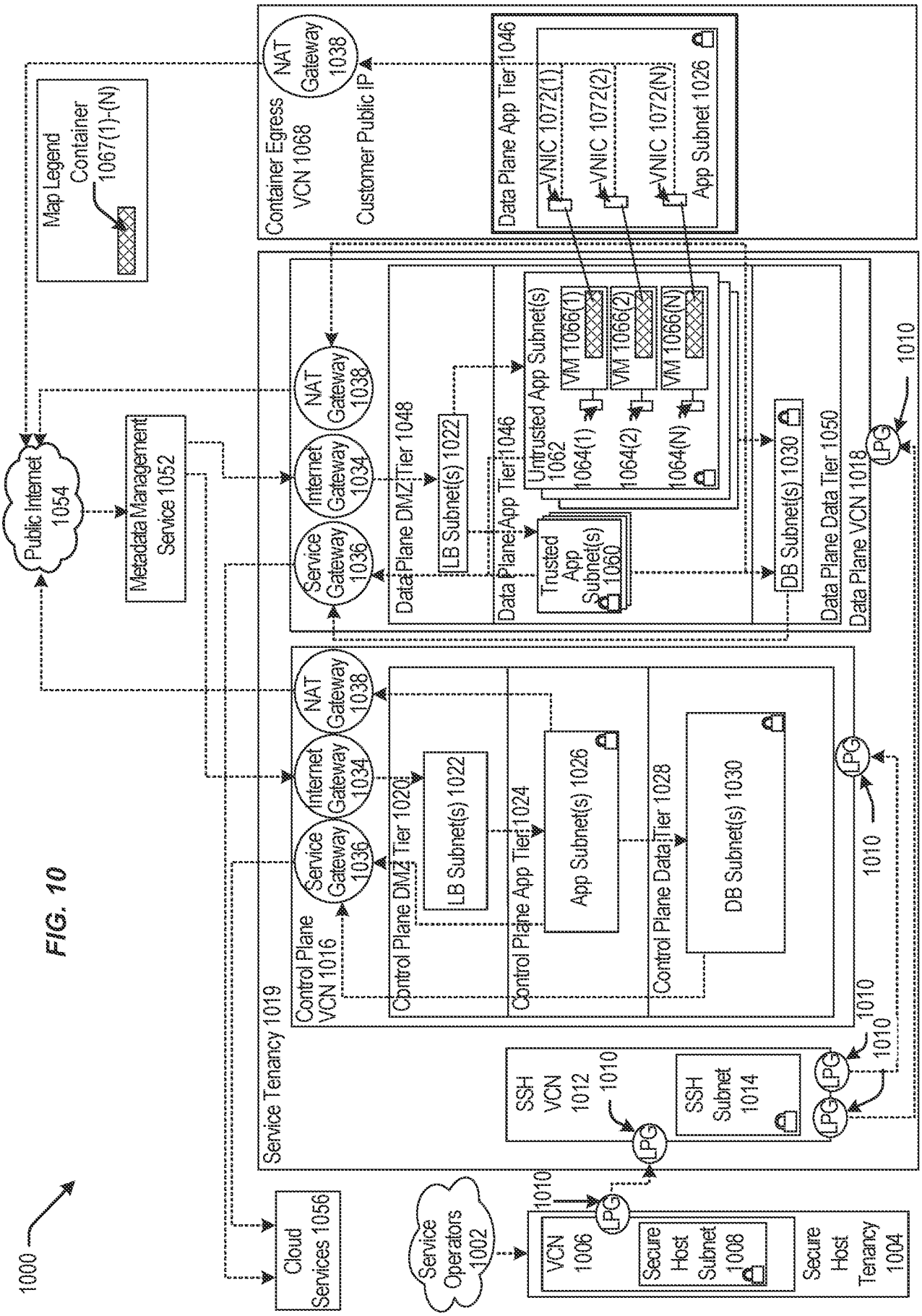

1000
Map Legend
Container 1067(1)-(N)
Public Internet 1054
Metadata Management Service 1052
Cloud Services 1056
Service Operators 1002
Container Egress VCN 1068
NAT Gateway 1038
Customer Public IP
Data Plane App Tier 1046
VNIC 1072(1)
VNIC 1072(2)
VNIC 1072(N)
App Subnet 1026
Service Tenancy 1019
Control Plane VCN 1016
Service Gateway 1036
Internet Gateway 1034
NAT Gateway 1038
Control Plane DMZ Tier 1020
LB Subnet(s) 1022
Control Plane App Tier 1024
App Subnet(s) 1026
Control Plane Data Tier 1028
DB Subnet(s) 1030
Data Plane DMZ Tier 1048
Data Plane App Tier 1046
Untrusted App Subnet(s) 1062
VM 1066(1)
VM 1066(2)
VM 1066(N)
1064(1)
1064(2)
1064(N)
Trusted App Subnet(s) 1060
Data Plane Data Tier 1050
Data Plane VCN 1018
1010
SSH VCN 1012
SSH Subnet 1014
VCN 1006
Secure Host Subnet 1008
Secure Host Tenancy 1004

FIG. 10

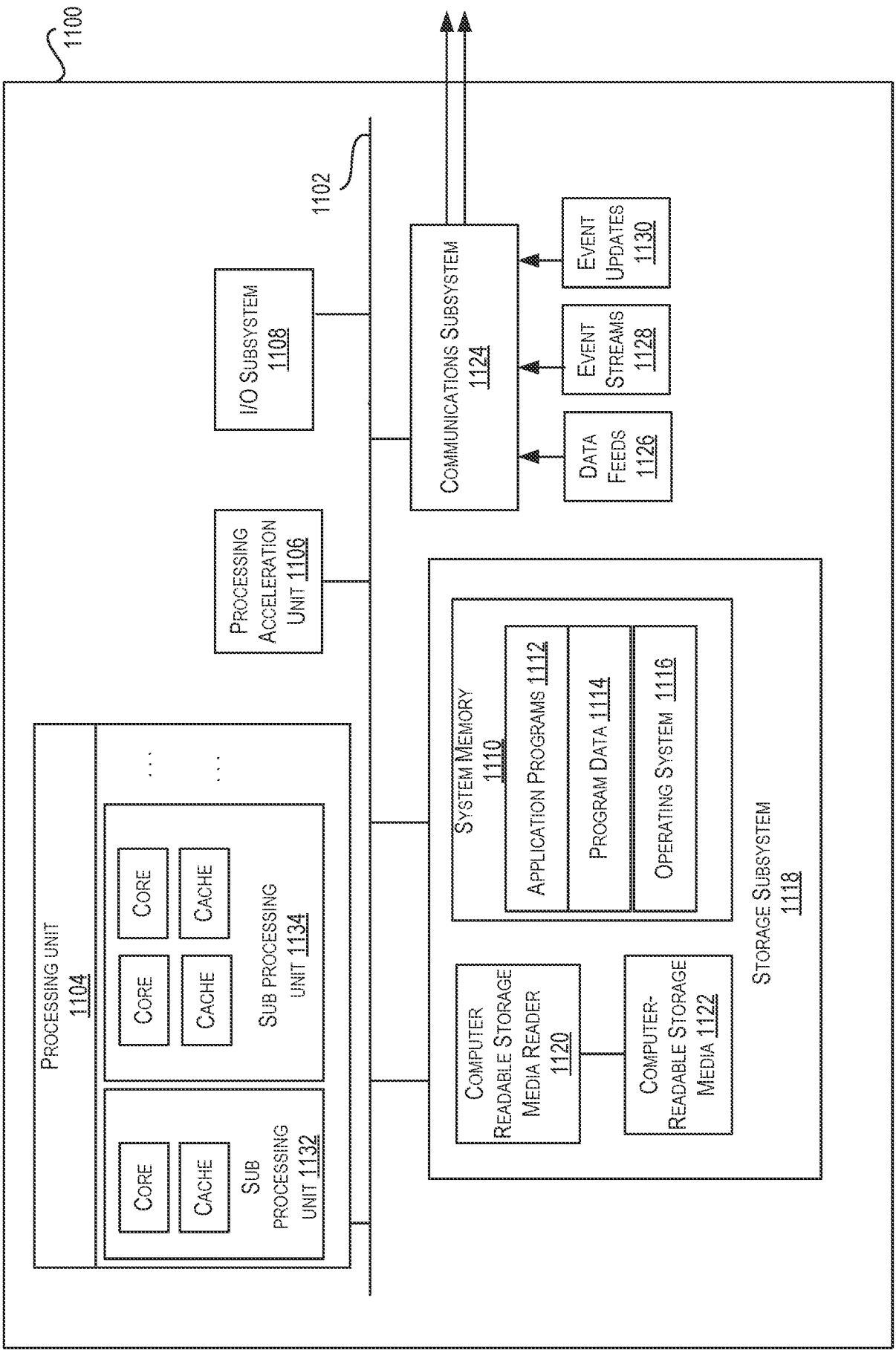

1100
1102
PROCESSING UNIT 1104
CORE
CACHE
SUB PROCESSING UNIT 1132
CORE
CACHE
SUB PROCESSING UNIT 1134
PROCESSING ACCELERATION UNIT 1106
I/O SUBSYSTEM 1108
COMMUNICATIONS SUBSYSTEM 1124
DATA FEEDS 1126
EVENT STREAMS 1128
EVENT UPDATES 1130
SYSTEM MEMORY 1110
APPLICATION PROGRAMS 1112
PROGRAM DATA 1114
OPERATING SYSTEM 1116
COMPUTER READABLE STORAGE MEDIA READER 1120
COMPUTER-READABLE STORAGE MEDIA 1122
STORAGE SUBSYSTEM 1118

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/062060

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/50 G06F8/60 G06F9/48
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2021/150307 A1 (ORACLE INT CORP [US]) 29 July 2021 (2021-07-29) the whole document	1-20
A	US 2017/228227 A1 (WINTERFELDT DAVID [US] ET AL) 10 August 2017 (2017-08-10) paragraphs [0028], [0086] - paragraph [0093]; figure 8B	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 April 2023	28/04/2023

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Renault, Sophie

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2023/062060

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2021150307 A1	29-07-2021	CN 114846447 A	02-08-2022
		EP 4094154 A1	30-11-2022
		JP 2023511113 A	16-03-2023
		WO 2021150307 A1	29-07-2021
US 2017228227 A1	10-08-2017	US 2013232480 A1	05-09-2013
		US 2016019096 A1	21-01-2016
		US 2017228227 A1	10-08-2017