

一种公有云的网络配置方法及相关设备

技术领域

5 本发明涉及云计算技术领域，尤其涉及一种公有云的网络配置方法及相关设备。

背景技术

云计算作为近年来新兴的产业，获得了科研界和产业界的广泛关注。云计算在全世界范围的兴起，以其灵活、高效、低成本、节能的运作方式成为推动产业绿色发展的重要引擎和 21 世纪新的商业平台。在公有云场景下，租户在公有云上部署虚拟机 (virtual machine, VM) 时，需要为虚拟机在租户的虚拟私有云 (virtual private cloud, VPC) 内的子网 (subnet) 内创建接口 (interface)，用以和 VPC 内其它的虚拟机进行通信。

15 现有公有云技术在虚拟机内支持容器，而容器之间的通信通过接口的子网际互连协议 (Sub IP) 支持，对每个虚拟机的接口，可以配置与接口 IP 相同网段的 Sub IP，每个 Sub IP 通过 IP 虚拟局域网 (virtual local area network, VLAN) 的方式在虚拟机内创建逻辑接口，并将逻辑接口与容器网络名空间关联。这种通信方式难以利用网络安全组实现容器通信安全且通信策略配置复杂，而且需要依赖 IP VLAN 模块将 Sub IP 转换为逻辑接口，容器地址与虚拟机的接口地址在同一个网段，容器发生迁移时可能将会导致地址修改，此外，由于弹性负载均衡 (elastic load balance, ELB)、弹性公网 IP (elastic IP, EIP) 等网络服务不支持 Sub IP 模型，因此，部署于虚拟机内的容器也不能支持 ELB、EIP 等网络服务。

发明内容

25 本发明实施例公开了一种公有云的网络配置方法及相关设备，能够对虚拟机的网口设置子接口，以实现虚拟机中的容器等通过设置的子接口使用公有云上的云服务，具备与虚拟机相同的网络能力。

30 第一方面，本申请提供一种公有云的网络配置方法，所述方法包括：为租户提供输入接口，所述输入接口用于接收所述租户输入的虚拟机网口标识，所述虚拟机网口标识用于识别需要配置子接口的虚拟机的网口；确定所述虚拟机网口标识对应的虚拟机的网口，为所述网口设置子接口，并将所述子接口的信息提供给所述租户。

在本申请提供的方案中，根据租户输入的虚拟机网口标识从而确定虚拟机的网口并为该网口配置子接口，然后将子接口的信息提供给租户，以使得虚拟机中的容器等可以通过配置的子接口使用公有云上的云服务，具备与虚拟机等同的网络能力。

35 结合第一方面，在第一方面的一种可能的实现方式中，所述输入接口还用于接收租户输入的所述子接口的配置信息，所述云管理平台根据所述配置信息配置所述子接口。

在本申请提供的方案中，租户通过输入子接口配置信息以使云管理平台配置子接口，保证配置得到的子接口具备与虚拟机相同的网络能力。

结合第一方面，在第一方面的一种可能的实现方式中，所述子接口与所述虚拟机中的容器绑定，所述容器通过所述子接口进行网络报文处理。

- 5 在本申请提供的方案中，通过将配置的子接口与虚拟机中的容器绑定，可以保证容器之间可以利用配置的子接口进行相互通信，从而完成网络报文处理，简化了容器通信过程。

- 10 结合第一方面，在第一方面的一种可能的实现方式中，所述子接口的配置信息包括安全组规则，所述安全组规则用于对绑定所述子接口的所述容器的报文进行安全组处理。

在本申请提供的方案中，可以利用网络安全组实现容器访问安全，简化了容器访问策略配置。

结合第一方面，在第一方面的一种可能的实现方式中，所述子接口的配置信息包括所述子接口的子网信息以及所述子接口的虚拟局域网标识 VlanID。

- 15 在本申请提供的方案中，利用子接口的子网信息和 VlanID 对子接口进行配置，保证所配置得到的子接口具备与虚拟机的主接口相同的网络能力。

结合第一方面，在第一方面的一种可能的实现方式中，确定所述虚拟机网口标识对应的虚拟机所在的服务器；通知所述服务器的虚拟机管理器在所述虚拟机的所述网口下设置所述子接口。

- 20 在本申请提供的方案中，网络配置器在接收到租户输入的虚拟机网口标识之后，首先确定该虚拟机网口标识对应的虚拟机所在的服务器，然后通知该服务器中的虚拟机管理器完成子接口的配置。

- 25 结合第一方面，在第一方面的一种可能的实现方式中，所述子接口的信息包括所述子接口的子接口标识 UUID、所述子接口的私网地址和子网信息、所述子接口的 MAC 地址以及所述子接口的 VlanID 中的一者或任意组合。

结合第一方面，在第一方面的一种可能的实现方式中，所述输入接口包括配置界面、应用程序编程接口 API 和输入模板。

- 30 在本申请提供的方案中，租户通过网络配置器提供的配置界面、API 和输入模板输入虚拟机网口标识和子接口配置信息，从而使得网络配置器能够完成子接口配置，使得配置得到的子接口具体与虚拟机的主接口相同的网络能力。

- 35 第二方面，本申请提供了一种网络设备，包括：接收模块，用于接收租户输入的虚拟机网口标识，所述虚拟机网口标识用于识别需要配置子接口的虚拟机的网口；处理模块，用于确定所述虚拟机网口标识对应的虚拟机的网口，为所述网口设置子接口，并将所述子接口的信息提供给所述租户。

结合第二方面，在第二方面的一种可能的实现方式中，所述接收模块，还用于接收所述租户输入的所述子接口的配置信息；所述处理模块，还用于根据所述配置信息配置所述子接口。

结合第二方面，在第二方面的一种可能的实现方式中，所述子接口与所述虚拟机

中的容器绑定，所述容器通过所述子接口进行网络报文处理。

结合第二方面，在第二方面的一种可能的实现方式中，所述子接口的配置信息包括安全组规则，所述安全组规则用于对绑定所述子接口的所述容器的报文进行安全组处理。

- 5 结合第二方面，在第二方面的一种可能的实现方式中，所述子接口的配置信息包括所述子接口的子网信息以及所述子接口的虚拟局域网标识 VlanID。

结合第二方面，在第二方面的一种可能的实现方式中，所述处理模块，具体用于：确定所述虚拟机网口标识对应的虚拟机所在的服务器；通知所述服务器的虚拟机管理器在所述虚拟机的所述网口下设置所述子接口。

- 10 结合第二方面，在第二方面的一种可能的实现方式中，所述子接口的信息包括所述子接口的子接口标识 UUID、所述子接口的私网地址和子网信息、所述子接口的媒体存取控制 MAC 地址以及所述子接口的虚拟局域网标识 VlanID 中的一者或任意组合。

结合第二方面，在第二方面的一种可能的实现方式中，所述输入接口包括配置界面、应用程序编程接口 API 和输入模板。

15

第三方面，本申请提供了一种计算设备，所述计算设备包括处理器和存储器，所述存储器用于存储程序代码，所述处理器用于调用所述存储器中的程序代码执行上述第一方面以及结合上述第一方面中的任意一种实现方式的方法。

- 20 第四方面，本申请提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，当该计算机程序被处理器执行时，可以实现上述第一方面以及结合上述第一方面中的任意一种实现方式所提供的公有云的网络配置方法的流程。

- 25 第五方面，本申请提供了一种计算机程序产品，该计算机程序产品包括指令，当该计算机程序被计算机执行时，使得计算机可以执行上述第一方面以及结合上述第一方面中的任意一种实现方式所提供的公有云的网络配置方法的流程。

附图说明

- 30 为了更清楚地说明本发明实施例技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

- 图 1 是本申请实施例提供的一种虚拟机中部署容器的场景示意图；
图 2 是本申请实施例提供的又一种虚拟机中部署容器的场景示意图；
35 图 3 是本申请实施例提供的一种系统架构的结构示意图；
图 4 是本申请实施例提供的一种公有云的网络配置方法的流程示意图；
图 5 是本申请实施例提供的一种配置界面的示意图；
图 6 是本申请实施例提供的一种设置子接口的场景示意图；
图 7 是本申请实施例提供的一种网络报文处理的示意图；

图 8 是本申请实施例提供的一种子接口的安全组配置示意图；

图 9 是本申请实施例提供的一种网络设备的结构示意图；

图 10 是本申请实施例提供的一种计算设备的结构示意图。

具体实施方式

5 下面结合附图对本申请实施例中的技术方案进行清楚、完整的描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

10 在本文中提及“实施例”意味着，结合实施例描述的特定特征、结构或特性可以包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例，也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是，本文所描述的实施例可以与其它实施例相结合。

15 首先，结合附图对本申请中所涉及的部分用语和相关技术进行解释说明，以便于本领域技术人员理解。

虚拟私有云（virtual private cloud, VPC）是一个公共云计算资源的动态配置池，需要使用加密协议、隧道协议和其它安全程序，在企业 and 云服务提供商之间传输数据。VPC 为弹性云服务器构建隔离的、用户自主配置和管理的虚拟网络环境，提升用户云中资源的安全性，简化用户的网络部署，可以在 VPC 中定义安全组、虚拟专用网（virtual private network, VPN）、IP 地址段、带宽等网络特性，租户可以通过 VPC 方便的管理、配置内部网络，进行安全快捷的网络变更。

20 网际协议虚拟局域网（IPVlan）可以从一个主机接口虚拟出多个虚拟网络子接口，每个子接口都具有相同的 MAC 地址，当使用动态主机配置协议（dynamic host configuration protocol, DHCP）分配 IP 时，需要配置唯一的主机标识（clientID）
25 字段作为机器的标识，IPVlan 有两种不同的工作模式，一个父接口（即主机接口）只能选择其中一种模式，不能采用混用模式，依附于它的所有虚拟子接口都会运行在这个模式下。

命名空间是 Linux 内核的一个强大特性，为容器虚拟化的实现带来极大便利，利用这个特性，每个容器都可以拥有自己单独的命名空间，运行在其中的应用都像是在
30 独立的操作系统环境中一样，命名机制保证了容器之间彼此互不影响，随着 Linux 系统对于命名空间功能的逐步完善，现在已经让进程在彼此隔离的命名空间中运行，虽然这些进程仍在共用同个内核和某些运行环境（例如一些系统命令和系统库），但是彼此之间是不可见的，并且认为自己是独占系统的。

安全组是一个逻辑上的分组，为同一个 VPC 内具有相同安全保护需求并相互信任
35 的弹性云服务器提供访问策略。安全组创建后，租户可以在安全组中定义各种访问规则，例如出方向规则和入方向规则，这些规则会对安全组内部的弹性云服务器出入方向网络流量进行访问控制，当弹性云服务器加入该安全组后，即受到这些访问规则的保护。

弹性网络接口（elastic network interface, ENI）是 VPC 中的一个逻辑网络组

件，代表虚拟网卡，ENI 是绑定私有网络内云服务器的一种弹性网络接口，可在多个云服务器间自由迁移，可以在云服务器上绑定多个网卡实现高可用网络或者在弹性网卡上绑定多个内网 IP，实现单主机多 IP 部署。

5 虚拟机 (virtual machine, VM) 是指通过软件模型的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。实体计算机中能够完成的工作在 VM 中都能够实现，在创建 VM 时，需要将实体计算机的部分硬盘和内存容量作为 VM 的硬盘和容量，每个 VM 都有独立的硬盘和操作系统等，可以像使用实体机一样对虚拟机进行操作。

10 容器 (container) 是计算机操作系统中的一种虚拟化技术，该技术使得进程运行于相对独立和隔离的环境 (包含独立的文件系统、命名空间、资源视图等)，从而能够简化软件的部署流程，增强软件的可移植性和安全性，提高系统资源的利用率，容器技术广泛应用于云计算领域的服务化场景。

网络：网络接口，例如为 eth0，eth0，网络接口在操作系统中表示一个网卡。

15 子接口：由网络接口虚拟出来的子接口，例如为 eth0.1，eth0.2，具体是从此网卡上虚拟出来的子网卡，子接口在操作系统中表示网卡虚拟出来的子网卡。

其中，上述的网卡可以是物理网卡，或者是分配给虚拟机使用的虚拟网卡。

在租户虚拟机中部署容器化应用的场景中，需要通过虚拟机接口的 SubIP 以支持容器。如图 1 所示，主机 100 和主机 200 之间通过虚拟机监控程序 (HyperVisor) 隧道进行连接，主机 100 中运行着虚拟机 110 和虚拟机监控程序 120，虚拟机 110 中部署了容器 1110 和容器 1120，虚拟机 110 的接口的 IP 为：10.1.0.100，为了支持容器 1110 和容器 1120，需要配置与虚拟机 110 接口 IP 相同网段的 SubIP，例如可以将容器 1110 对应的 IP 配置为：10.1.0.101，将容器 1120 对应的 IP 配置为：10.1.0.102，在配置 SubIP 完成之后，每个 SubIP 通过 IPvlan 方式在虚拟机 110 内创建逻辑接口，20 并将创建的逻辑接口与容器命名空间关联，以使得每个容器都对应一个逻辑接口。同理，主机 200 与主机 100 类似，通过上述相同的方法为虚拟机 210 中部署的容器 2110 和容器 2120 配置 SubIP 并创建相应的逻辑接口。在完成上述配置之后，容器之间就可以正常进行通信了，例如虚拟机 110 中的容器 1110 可以与虚拟机 210 中的容器 2110 进行通信。

30 可以看出，在图 1 所示的利用 SubIP 支持容器的场景中，需要将 SubIP 转换为逻辑接口才能绑定容器命名空间，且容器地址与虚拟机地址在同一个网段，若容器迁移至别的虚拟机，与原来虚拟机地址不在同一个网段时，需要修改容器地址。此外，由于网络安全组采用端口模型，因此不能利用一个安全组来表达一个容器应用，无法利用网络安全组实现容器安全，导致容器访问策略配置复杂，且 ELB、EIP 等网络服务不支持 SubIP，所以容器也无法支持 ELB、EIP 等网络服务。

35 或者，通过 VPC 内自定义路由表以支持容器。如图 2 所示，主机 300 和主机 400 之间通过 HyperVisor 隧道连接，它们属于同一个 VPC，主机 300 中运行着虚拟机 310 和虚拟机监控程序 320，虚拟机 310 中部署了容器 3110、容器 3120 和网关 3130，虚拟机 310 的接口 IP 为：10.1.0.100，容器 3110 和容器 3120 使用一个固定的 IP 地址

段,例如容器 3110 对应的 IP 为:192.168.1.2,容器 3120 对应的 IP 为:192.168.1.3,针对每个容器对应的 IP,在虚拟机 310 内创建逻辑接口,并将创建的逻辑接口与容器名空间关联,以使得每个容器都对应一个逻辑接口。同理,主机 400 中运行的虚拟机 410 中所部署的容器 4110 和容器 4120 也使用一个固定的 IP 地址段,例如容器 4110 对应的 IP 为:192.168.2.2,容器 4120 对应的 IP 为:192.168.2.3,在虚拟机 410 内创建逻辑接口,并与容器名空间关联。之后,将 IP 地址段配置为 VPC 的路由表,示例性的,192.168.1.0 至 192.168.1.16 的地址段对应虚拟机 310,192.168.2.0 至 192.168.2.16 的地址段对应虚拟机 410,然后将配置得到的路由表保存至每个虚拟机的网关中(例如网关 3130 和网关 4130),这样,不同容器之间可以进行正常通信,当容器需要访问另一个虚拟机中的容器时,可以利用 VPC 自定义路由表的表项进行访问。

同样的,图 2 所示的利用 VPC 自定义路由表支持容器的场景中,也需要将容器对应的 IP 转换为逻辑接口才能绑定容器名空间,不同的虚拟机使用独立的网段,容器跨虚拟机迁移后需要修改容器地址,每个虚拟机内需要做转发网关以处理默认路由、地址解析协议(address resolution protocol, ARP)等,导致容器实现复杂。此外,也无法利用网络安全组实现容器安全,不能支持 ELB、EIP 等网络服务,也不能用于访问对象存储服务(object storage service, OBS)等云服务。

为了解决上述问题,本申请提供了一种公有云的网络配置方法及相关设备,该方法应用于云管理平台,通过租户输入的虚拟机网口标识,识别出需要配置子接口的虚拟机的网口,并对该网口进行设置得到子接口,利用设置得到的子接口以支持该虚拟机中的容器等应用程序,使得容器具备和虚拟机一样的网络能力,且可以利用网络安全组实现容器安全并支持各种网络服务。

本申请实施例的技术方案还可以应用于各种需要依赖于公有云下网络子接口的场景,包括但不限于网络功能虚拟化(network function virtualization, NFV)场景、虚拟机支持多租户、多 VPC 访问场景等。

图 3 是本申请实施例提供的一种系统架构的示意图。如图 3 所示,该系统包括虚拟机 310 和虚拟机 320,虚拟机 310 存在两个以太网口,分别为网口 3110 和网口 3120,其中针对网口 3110,设置了两个子接口,即子接口 3111 和子接口 3112,针对网口 3120 设置了子接口 3121,虚拟机 320 中存在一个以太网口 3210,针对该网口设置了两个子接口,分别为子接口 3211 和子接口 3212。需要说明的是,各个子接口在虚拟机内部都是 ENI 的一个子接口,每个子接口对应唯一的 VlanID,每个子接口都可以绑定与网口不同的子网,子网可以属于不同的 VPC。例如,网口 3110、子接口 3111、子接口 3112 以及子接口 3211 都属于子网 1,网口 3120 和网口 3210 属于子网 2,子接口 3121 和子接口 3212 属于子网 3,其中,子网 1 和子网 2 属于同一个 VPC,即 VPC1,子网 3 属于 VPC2。

基于上述,下面对本申请实施例提供的公有云的网络配置方法及相关设备进行描述。参见图 4,图 4 为本申请实施例提供的一种公有云的网络配置方法的流程示意图。

如图 4 所示, 该方法包括但不限于以下步骤:

S401: 网络配置器为租户提供输入接口。

具体地, 网络配置器向租户提供输入接口以接收租户输入的虚拟机网口标识, 由于虚拟机网口标识是唯一的, 所以网络配置器可以根据租户输入的虚拟机网口标识在公有云数据中心中选择该虚拟机网口标识对应的虚拟机, 并确定在该虚拟机中需要配置的网口。示例性的, 该虚拟机网口标识可以是上述图 3 中的网口 3110。

可选的, 网络配置器所提供的输入接口可以包括配置界面、API 接口和输入模板, 网络配置器可以任意选择一种方式以接收租户输入的虚拟机网口标识, 本申请对此不作限定。

10 S402: 网络配置器接收租户输入的子接口的配置信息。

具体地, 网络配置器在确定需要配置的网口之后, 进一步接收租户输入的针对该网口对应的子接口的配置信息, 以使得云管理平台可以根据接收到的子接口的配置信息配置子接口。

该子接口的配置信息可以包括子接口的子网信息以及子接口的 VlanID, 在获取到该子接口的子网信息以及 VlanID 之后可以配置子接口。可以理解, 通过为子接口配置子网和 VlanID, 可以保证配置得到的子接口与虚拟机的网口具备相同的网络能力, 能够支持各种网络服务。

示例性的, 以网络配置器所提供的输入接口为配置界面为例, 如图 5 所示, 租户在该配置界面中输入虚拟机网口标识以及子接口的子网信息和 VlanID, 租户在输入完成相关参数配置之后, 点击界面中的确认按钮, 云管理平台即可以根据租户输入的配置信息配置子接口。

S403: 网络配置器通知虚拟机管理器设置子接口。

具体地, 网络配置器根据租户输入的虚拟机网口标识确定该虚拟机网口标识对应的虚拟机所在的服务器, 然后通知该服务器的虚拟机管理器在所述虚拟机的网口下设置子接口。

进一步的, 虚拟机管理器在虚拟机内部创建容器对应的 SubENI, 然后创建对应的 VLAN 子接口, 并将该子接口划归对应的容器网络名空间, 此时该子接口将通过动态主机配置协议 (dynamic host configuration protocol, DHCP) 自动获取 IP, 以便于后续相互进行通信。

需要说明的是, SubENI 是系统中的网络接口, 其 API 模型与已有的 ENI 的 API 模型一致, 对于其它使用网络接口的网络服务来说, 其感知不到 SubENI 与 ENI 的区别。由于 SubENI 的 API 模型与 ENI 的 API 模型一致, 所以在创建 SubENI 时, 对于 SubENI 的 API 接口是直接 ENI 的 API 上进行扩展, 但是增加了三个属性, 分别为: 接口类型、主接口 ID 和子接口对应的 VLAN, 其中, 接口类型是用字符串进行表示的, 包含 ENI 主接口和 SubENI 子接口; 主接口 ID 也是用字符串进行表示的, 是在接口类型为 SubENI 子接口时有效, 表示主接口的 ID; 子接口对应的 VLAN 是用整数进行表示的, 其内容为子接口的 VlanID, 该值可以由虚拟机管理器创建 SubENI 时在主接口的 VLAN 空间中自动分配可用 VlanID, 或者是由租户指定, 但必须保证所指定的值在主接口的 VLAN 空间中不发生冲突, 该值的取值范围是 1-4094。

示例性的，参见图 6，图 6 是本申请实施例提供的一种设置子接口场景示意图。如图 6 所示，虚拟机 610、虚拟机 620 和虚拟机管理器 630 属于公有云上的同一个 VPC，它们之间通过 ENI 进行通信，虚拟机 610 中部署了网口 6110、网口 6120、节点代理（kubelet）6130、容器 6140、容器 6150、容器 6160 和容器 6170，虚拟机 620 中部署了网口 6210、节点代理 6220、容器 6230 和容器 6240，虚拟机管理器 630 包含网口 6310 和管理程序 6320，管理程序 6320 通过公有云提供的 API 网关调用公有云的 API，在虚拟机 610 中创建容器对应的 SubENI，在创建 SubENI 时可以指定 VLAN 或者让公有云自行分配 VLAN，然后通过节点代理 6130 创建 SubENI 对应的 VLAN 子接口，即子接口 6111、子接口 6112、子接口 6113 和子接口 6114，并将子接口划归对应的容器网络名空间，例如将子接口 6111 划归为容器 6140，将子接口 6112 划归为容器 6150，将子接口 6113 划归为容器 6160，将子接口 6114 划归为容器 6170；以及在虚拟机 620 中创建容器对应的 SubENI，然后通过节点代理 6220 创建 SubENI 对应的子接口 6211 和子接口 6212，并将子接口 6211 划归为容器 6230，将子接口 6212 划归为容器 6240。在将子接口划归对应的容器网络名空间之后，各个子接口将通过 DHCP 自动获取 IP，此时，各个容器之间可以进行通信以实现业务互通，且各个容器能够支持各种网络服务，具备与虚拟机相同的网络能力。

在一种可能的实现方式中，所述子接口与虚拟机中的容器绑定，所述容器通过所述子接口进行网络报文处理。

具体地，由于子接口具备子网信息和 VlanID，因此在将子接口在与虚拟机中的容器绑定之后，容器具备和虚拟机一样的网络能力，可以通过子接口进行网络报文处理。

示例性的，如图 7 所示，主机 710 中部署了虚拟机 7110 和虚拟交换机 7120，虚拟机 7110 中部署了容器 71110 和容器 71120，容器 71110 与子接口 71111 绑定，容器 71120 与子接口 71121 绑定，主机 720 的结构与主机 710 类似，其同样部署了虚拟机 7210 和虚拟交换机 7220，虚拟机 7210 中部署了容器 72110 和容器 72120，容器 72110 与子接口 72111 绑定，容器 72120 与子接口 72121 绑定。若容器 71110 需要与容器 72110 进行通信，那么容器 71110 通过子接口 71111 发送的报文是一个携带 VlanID 的报文，即报文 1，该报文 1 包括 6 比特的目的地址、6 比特的源地址、4 比特的虚拟局域网标识（VlanID）、2 比特的报文类型和有效载荷，虚拟交换机 7120 在接收到报文 1 之后，根据发送该报文的子接口 71111 的 VlanID，找到子接口 71111 对应的路由表（即 VPC 路由表），之后根据路由表中的路由对报文 1 进行 VxLAN 封装，虚拟交换机 7120 删掉报文 1 的虚拟局域网标识字段，并使用目的接口信息（即子接口 72111）进行 VxLAN 封装得到报文 2，报文 2 与报文 1 相比，在其头部增加了隧道端点字段、8 比特的用户数据报协议（user datagram protocol，UDP）字段和目的接口字段（例如目的接口所属的子网或者目的接口所属的 VPC），去掉了虚拟局域网标识字段。虚拟交换机 7120 将报文 2 发送至虚拟交换机 7220，虚拟交换机 7220 在接收到报文 2 之后，根据报文 2 中的目的接口字段找到对应的路由表（即 VPC 路由表），然后去掉报文 2 的隧道端点字段、UDP 字段和目的接口字段，之后根据报文中的目的地址查表转发，确定是虚拟机 7210 中的子接口 72111，添加虚拟局域网标识字段，应理解，这里添加的虚拟局域网标识与报文 1 中的虚拟局域网标识是不一样的，得到报文 3，最后将报文 3 发送给

虚拟机 7210, 虚拟机 7210 在接收到报文 3 之后, 将去掉其中的虚拟局域网标识字段, 然后通过子接口 72111 发送给容器 72110, 最终成功实现容器 71110 和容器 72110 之间的通信。

5 在一种可能的实现方式中, 所述子接口的配置信息还包括安全组规则, 该安全组规则用于对绑定子接口的容器的报文进行安全组处理。

具体地, 由于配置得到的子接口具备和虚拟机的主接口一样的网络能力, 因此可以利用一个安全组来表达一个容器应用, 然后对绑定子接口的容器报文进行安全组处理, 保证容器应用访问安全。

进一步的, 一个应用使用一个安全组, 安全组内部安全可信, 不同的应用之间可以通过安全组之间的安全策略配置访问策略。示例性的, 如图 8 所示, 虚拟机 810 中部署了容器 8110、容器 8120、容器 8130 和容器 8140, 其中容器 8110 与子接口 8111 进行绑定, 容器 8120 与子接口 8121 进行绑定, 容器 8130 与子接口 8131 进行绑定, 容器 8140 与子接口 8141 进行绑定; 虚拟机 820 中部署了容器 8210 和容器 8220, 容器 8210 与子接口 8211 进行绑定, 容器 8220 与子接口 8221 进行绑定。将子接口 8111 和子接口 8131 配置到安全组 1 中, 将子接口 8121 和子接口 8211 配置到安全组 2 中, 将子接口 8141 和子接口 8221 配置到安全组 3 中, 同一个安全组中的子接口可以互相访问, 不受限制, 例如子接口 8111 可以直接访问子接口 8131, 子接口 8141 可以直接访问子接口 8221, 不同安全组之间可以使用默认安全组规则或者根据需要设置安全组规则, 以满足实际需要和保证容器应用访问安全。例如, 若将安全组规则设置为安全组 1 允许访问安全组 2, 安全组 2 不允许访问安全组 1, 那么子接口 8111 可以访问子接口 8121 和子接口 8211, 子接口 81221 和子接口 8211 不能访问子接口 8111 和子接口 8131。

此外, 将子接口配置到安全组中, 可以保证在容器发生迁移时, 其所属的安全组不会发生改变, 容器之间的访问仍遵循安全组规则, 不需要修改容器地址以及额外的进行相关访问策略配置。例如, 将容器 8110 迁移至虚拟机 820 中, 那么子接口 8111 仍属于安全组 1。

S404: 网络配置器将子接口的信息提供给租户。

具体地, 在虚拟机管理器完成子接口配置之后, 将得到子接口的相关信息, 例如子接口的子接口标识、子接口的私网地址和子网信息、子接口的 MAC 地址和子接口的 VlanID, 然后将子接口的信息提供给租户以使得租户可以为子接口配置各种网络服务。

S405: 网络配置器通过子接口给容器配置网络服务。

具体地, 子接口在与容器绑定之后, 此时容器的网络能力与虚拟机的网络能力是一样的, 所有虚拟机能够支持的网络服务, 容器同样能够支持, 例如挂载 ELB、绑定 EIP、访问 OBS 等, 这样可以扩展容器的适用场景。

35

上述详细阐述了本申请实施例的方法, 为了便于更好的实施本申请实施例的上述方案, 相应地, 下面还提供用于配合实施上述方案的相关设备。

参见图 9, 图 9 是本申请实施例提供的一种网络设备的结构示意图, 该网络设备可以是上述图 4 所述的方法实施例中的网络配置器, 可以执行图 4 所述的公有云的网

网络配置方法实施例中以网络配置器为执行主体的方法和步骤。如图 9 所示，该网络设备 900 包括接收模块 910 和处理模块 920。其中，

接收模块 910，用于接收租户输入的虚拟机网口标识，所述虚拟机网口标识用于识别需要配置子接口的虚拟机的网口；

- 5 处理模块 920，用于确定所述虚拟机网口标识对应的虚拟机的网口，为所述网口设置子接口，并将所述子接口的信息提供给所述租户。

具体地，所述接收模块 910 用于执行前述步骤 S401 和 S402，且可选的执行前述步骤中可选的方法，所述处理模块 920 用于执行前述步骤 S403-S405，且可选的执行前述步骤中可选的方法。上述两个模块之间互相可以通过通信通路进行数据传输，应
10 理解，网络设备 900 包括的各模块可以软件模块、也可以为硬件模块、或部分为软件模块部分为硬件模块。

作为一个实施例，所述接收模块 910，还用于接收所述租户输入的所述子接口的配置信息；所述处理模块 920，还用于根据所述配置信息配置所述子接口。

- 15 作为一个实施例，所述子接口与所述虚拟机中的容器绑定，所述容器通过所述子接口进行网络报文处理。

作为一个实施例，所述子接口的配置信息包括安全组规则，所述安全组规则用于对绑定所述子接口的所述容器的报文进行安全组处理。

作为一个实施例，所述子接口的配置信息包括所述子接口的子网信息以及所述子接口的虚拟局域网标识 VlanID。

- 20 作为一个实施例，所述处理模块 920，具体用于：确定所述虚拟机网口标识对应的虚拟机所在的服务器；通知所述服务器的虚拟机管理器在所述虚拟机的所述网口下设置所述子接口。

作为一个实施例，所述子接口的信息包括所述子接口的子接口标识 UUID、所述子接口的私网地址和子网信息、所述子接口的媒体存取控制 MAC 地址以及所述子接口的
25 虚拟局域网标识 VlanID 中的一者或任意组合。

作为一个实施例，所述输入接口包括配置界面、应用程序编程接口 API 和输入模板。

可以理解，本申请实施例中的接收模块 910 可以由收发器或收发器相关电路组件实现，处理模块 920 可以由处理器或处理器相关电路组件实现。

- 30 需要说明的是，上述网络设备的结构仅仅作为一种示例，不应构成具体限定，可以根据需要对该网络设备中的各个模块进行增加、减少或合并。此外，该网络设备中的各个模块的操作和/或功能是为了实现上述图 4 所描述的方法的相应流程，为了简洁，在此不再赘述。

- 35 参见图 10，图 10 是本申请实施例提供的一种计算设备的结构示意图。如图 10 所示，该计算设备 10 包括：处理器 11、通信接口 12 以及存储器 13，所述处理器 11、通信接口 12 以及存储器 13 通过内部总线 14 相互连接。应理解，该计算设备 10 可以是公有云中的计算设备。

所述处理器 11 可以由一个或者多个通用处理器构成，例如中央处理器（central

processing unit, CPU), 或者 CPU 和硬件芯片的组合。上述硬件芯片可以是专用集成电路(application-specific integrated circuit, ASIC)、可编程逻辑器件(programmable logic device, PLD) 或其组合。上述 PLD 可以是复杂可编程逻辑器件(complex programmable logic device, CPLD)、现场可编程逻辑门阵列

5 (field-programmable gate array, FPGA)、通用阵列逻辑(generic array logic, GAL) 或其任意组合。

总线 14 可以是外设部件互连标准(peripheral component interconnect, PCI) 总线或扩展工业标准结构(extended industry standard architecture, EISA) 总线等。所述总线 14 可以分为地址总线、数据总线、控制总线等。为便于表示, 图 10 中
10 仅用一条粗线表示, 但不表示仅有一根总线或一种类型的总线。

存储器 13 可以包括易失性存储器(volatile memory), 例如随机存取存储器(random access memory, RAM); 存储器 730 也可以包括非易失性存储器(non-volatile memory), 例如只读存储器(read-only memory, ROM)、快闪存储器(flash memory)、硬盘(hard disk drive, HDD) 或固态硬盘(solid-state drive, SSD); 存储器 13
15 还可以包括上述种类的组合。

需要说明的是, 计算设备 10 的存储器 13 中存储了网络设备 900 的各个模块对应的代码, 处理器 11 执行这些代码实现了网络设备 900 的各个模块的功能, 即执行了 S401-S405 的方法。

本申请还提供一种计算机可读存储介质, 其中, 所述计算机可读存储介质存储有
20 计算机程序, 当该计算机程序被处理器执行时, 可以实现上述方法实施例中记载的任意一种的部分或全部步骤。

本发明实施例还提供一种计算机程序, 该计算机程序包括指令, 当该计算机程序被计算机执行时, 使得计算机可以执行任意一种发放区域资源的方法的部分或全部步骤。

25 在上述实施例中, 对各个实施例的描述都各有侧重, 某个实施例中未详述的部分, 可以参见其它实施例的相关描述。

需要说明的是, 对于前述的各方法实施例, 为了简单描述, 故将其都表述为一系列的动作组合, 但是本领域技术人员应该知悉, 本申请并不受所描述的动作顺序的限制, 因为依据本申请, 某些步骤可能可以采用其它顺序或者同时进行。其次, 本领域
30 技术人员也应该知悉, 说明书中所描述的实施例均属于优选实施例, 所涉及的动作和模块并不一定是本申请所必须的。

在本申请所提供的几个实施例中, 应该理解到, 所揭露的装置, 可通过其它的方式实现。例如, 以上所描述的装置实施例仅仅是示意性的, 例如上述单元的划分, 仅仅为一种逻辑功能划分, 实际实现时可以有另外的划分方式, 例如多个单元或组件可以结合或者可以集成到另一个系统, 或一些特征可以忽略, 或不执行。另一点, 所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口, 装置或单元的间接耦合或通信连接, 可以是电性或其它的形式。

上述作为分离部件说明的单元可以是或者也可以不是物理上分开的, 作为单元显示的部件可以是或者也可以不是物理单元, 即可以位于一个地方, 或者也可以分布到

多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的

5 单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

权 利 要 求 书

- 1、一种公有云的网络配置方法，其特征在于，所述方法应用于云管理平台，包括：
为租户提供输入接口，所述输入接口用于接收所述租户输入的虚拟机网口标识，
所述虚拟机网口标识用于识别需要配置子接口的虚拟机的网口；
- 5 确定所述虚拟机网口标识对应的虚拟机的网口，为所述网口设置子接口，并将所述子接口的信息提供给所述租户。
- 2、如权利要求1所述的方法，其特征在于，所述输入接口还用于接收所述租户输入的所述子接口的配置信息，所述方法还包括：
所述云管理平台根据所述配置信息配置所述子接口。
- 10 3、如权利要求1或2所述的方法，其特征在于，所述子接口与所述虚拟机中的容器绑定，所述容器通过所述子接口进行网络报文处理。
- 4、如权利要求3所述的方法，其特征在于，所述子接口的配置信息包括安全组规则，所述安全组规则用于对绑定所述子接口的所述容器的报文进行安全组处理。
- 5、如权利要求1至4任一项所述的方法，其特征在于，所述子接口的配置信息包括所述子接口的子网信息以及所述子接口的虚拟局域网标识VlanID。
- 15 6、如权利要求4或5所述的方法，其特征在于，所述确定所述虚拟机网口标识对应的虚拟机的网口，为所述网口设置子接口，并将所述子接口的信息提供给所述租户，包括：
确定所述虚拟机网口标识对应的虚拟机所在的服务器；
- 20 通知所述服务器的虚拟机管理器在所述虚拟机的所述网口下设置所述子接口。
- 7、如权利要求1至6任一项所述的方法，其特征在于，所述子接口的信息包括所述子接口的子接口标识UUID、所述子接口的私网地址和子网信息、所述子接口的媒体存取控制MAC地址以及所述子接口的虚拟局域网标识VlanID中的一者或任意组合。
- 8、如权利要求1至7任一项所述的方法，其特征在于，所述输入接口包括配置界面、应用程序编程接口API和输入模板。
- 25 9、一种网络设备，其特征在于，包括：
接收模块，用于接收租户输入的虚拟机网口标识，所述虚拟机网口标识用于识别需要配置子接口的虚拟机的网口；
处理模块，用于确定所述虚拟机网口标识对应的虚拟机的网口，为所述网口设置
- 30 子接口，并将所述子接口的信息提供给所述租户。
- 10、如权利要求9所述的网络设备，其特征在于，
所述接收模块，还用于接收所述租户输入的所述子接口的配置信息；
所述处理模块，还用于根据所述配置信息配置所述子接口。
- 11、如权利要求9或10所述的网络设备，其特征在于，所述子接口与所述虚拟机
- 35 中的容器绑定，所述容器通过所述子接口进行网络报文处理。
- 12、如权利要求11所述的网络设备，其特征在于，所述子接口的配置信息包括安全组规则，所述安全组规则用于对绑定所述子接口的所述容器的报文进行安全组处理。
- 13、如权利要求9至12任一项所述的网络设备，其特征在于，所述子接口的配置信息包括所述子接口的子网信息以及所述子接口的虚拟局域网标识VlanID。

14、如权利要求 12 或 13 所述的网络设备，其特征在于，所述处理模块，具体用于：

确定所述虚拟机网口标识对应的虚拟机所在的服务器；

通知所述服务器的虚拟机管理器在所述虚拟机的所述网口下设置所述子接口。

5 15、如权利要求 9 至 14 任一项所述的网络设备，其特征在于，所述子接口的信息包括所述子接口的子接口标识 UUID、所述子接口的私网地址和子网信息、所述子接口的媒体存取控制 MAC 地址以及所述子接口的虚拟局域网标识 VlanID 中的一者或任意组合。

10 16、如权利要求 9 至 15 任一项所述的网络设备，其特征在于，所述输入接口包括配置界面、应用程序编程接口 API 和输入模板。

17、一种计算设备，其特征在于，所述计算设备包括存储器和处理器，所述处理器执行存储器存储的计算机指令，使得所述计算设备执行权利要求 1-8 任一项所述的方法。

15 18、一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，当所述计算机程序被处理器执行时，所述处理器执行权利要求 1-8 任一项所述的方法。