

NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, 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)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

自动搭建Kubernetes主节点的方法、装置、终端设备及 可读存储介质

[0001] 本申请要求于2018年03月31日提交中国专利局、申请号为201810276602.8、发明名称为“自动搭建Kubernetes主节点的方法及终端设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

[0002] 本申请属于数据处理技术领域，尤其涉及一种自动搭建Kubernetes主节点的方法、装置、终端设备及计算机可读存储介质。

背景技术

[0003] 传统的虚拟化技术，如hypervisor虚拟化技术在性能和资源使用率等方面存在不足，而容器（container）技术通过将单个操作系统管理的资源划分到孤立的组中，提升了资源使用率，逐渐成为研究热门。容器可理解为一种沙盒。每个容器内运行一个应用，不同的容器相互隔离，但可在容器之间建立通信机制。Docker容器技术允许在同一台主机或虚拟机上运行若干个容器，每个容器为独立的虚拟环境或应用。

[0004] 通常来说，需要使用工具对容器进行控制。Kubernetes为一款开源的容器操作平台，其可以实现将若干个容器组合成一个服务及动态地分配容器运行的主机等功能，为用户使用容器提供了极大的便利。通常来说，Kubernetes集群包括主节点和从节点两类节点，其中主节点负责对Kubernetes集群中所有资源进行管控和调度，故要使用Kubernetes，首先需要进行主节点的搭建。但在现有技术中，主要通过用户手动搭建Kubernetes主节点，并且搭建过程依赖互联网服务，搭建效率和适用性低。

发明概述

技术问题

[0005] 有鉴于此，本申请实施例提供了一种自动搭建Kubernetes主节点的方法、装置、终端设备及计算机可读存储介质，以解决现有技术中需要依靠手动搭建Kuber

netes主节点，搭建效率和适用性低的问题。

问题的解决方案

技术解决方案

[0006] 本申请实施例的第一方面提供了一种自动搭建Kubernetes主节点的方法，包括：

[0007] 获取预设的用于搭建Kubernetes主节点的脚本文件；

[0008] 执行所述脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数，生成配置文件；

[0009] 启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点。

[0010] 本申请实施例的第二方面提供了一种自动搭建Kubernetes主节点的装置，可以包括用于实现上述自动搭建Kubernetes主节点的方法的步骤的单元。

[0011] 本申请实施例的第三方面提供了一种终端设备，包括存储器以及处理器，所述存储器中存储有可在所述处理器上运行的计算机可读指令，所述处理器执行所述计算机可读指令时实现上述自动搭建Kubernetes主节点的方法的步骤。

[0012] 本申请实施例的第四方面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可读指令，所述计算机可读指令被处理器执行时实现上述自动搭建Kubernetes主节点的方法的步骤。

发明的有益效果

有益效果

[0013] 本申请实施例通过获取预设的用于搭建Kubernetes主节点的脚本文件，并通过执行预设的脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数生成配置文件，最后启动SaltStack工具，在SaltStack工具控制下，自动根据该配置文件执行对Kubernetes主节点的搭建，实现了Kubernetes主节点的自动搭建，并且不需依赖互联网服务，提升了搭建效率和对搭建环境的适用性。

对附图的简要说明

附图说明

[0014] 图1是本申请实施例一中自动搭建Kubernetes主节点的方法的实现流程图；

- [0015] 图2是本申请实施例二中自动搭建Kubernetes主节点的方法的实现流程图；
- [0016] 图3是本申请实施例三中自动搭建Kubernetes主节点的方法的实现流程图；
- [0017] 图4是本申请实施例四中自动搭建Kubernetes主节点的方法的实现流程图；
- [0018] 图5是本申请实施例五中自动搭建Kubernetes主节点的方法的实现流程图；
- [0019] 图6是本申请实施例六中自动搭建Kubernetes主节点的装置的结构框图；
- [0020] 图7是本申请实施例七中一个Kubernetes集群的示意图；
- [0021] 图8是本申请实施例八中基于Kubernetes集群的区域结构图；
- [0022] 图9是本申请实施例九中终端设备的示意图。

发明实施例

本发明的实施方式

- [0023] 为了对本申请的技术特征、目的和效果有更加清楚的理解，现对照附图详细说明本申请的具体实施方式。
- [0024] 请参阅图1，图1是本申请实施例提供的一种自动搭建Kubernetes主节点的方法的实现流程图。如图1所示，该方法包括以下步骤：
- [0025] S101：获取预设的用于搭建Kubernetes主节点的脚本文件。
- [0026] 在本申请实施例中，主要通过SaltStack工具进行Kubernetes主节点的搭建，首先，获取预设的用于搭建Kubernetes主节点的脚本文件，为了便于说明，对Kubernetes及Kubernetes主节点进行介绍。Kubernetes为一款自动化容器操作的开源平台，能够实现对容器的部署、调度和节点集群间扩展等功能，将安装有Kubernetes平台的物理服务器或虚拟机称作Kubernetes节点。通常来说，需要多个Kubernetes节点组建为Kubernetes集群（Kubernetes Cluster）以实现对容器的部署和管理。在一个Kubernetes集群内，有且只有一套控制单元，即Kubernetes主节点（Kubernetes Master），主要负责接收请求、资源调度以及进行管理，Kubernetes主节点的组成部分在后文进行具体阐述。除了Kubernetes主节点之外，Kubernetes集群内还包括多个Kubernetes主节点的下属对象，即Kubernetes从节点（Kubernetes Node），用于实际运行由Kubernetes主节点分配的容器。
- [0027] 对于搭建过程，相比于传统的手动编写Kubernetes主节点的每一条配置代码，在本申请实施例中，主要通过脚本文件搭建Kubernetes主节点，脚本文件是确定

的一系列控制计算机进行运算操作动作的组合，在其中可实现逻辑分支，脚本文件可采用Shell语言等编写。

[0028] 可选地，在接收到用户关于Kubernetes主节点的搭建指令后，获取文件服务器PackageServer上存储的脚本文件。在本申请实施例中，可向用户提供搭建Kubernetes主节点的搭建选项，搭建选项可以以图形化界面或者命令行的形式呈现至用户，在获取到用户关于该搭建选项的确认结果，即获取到用户的搭建指令后，执行获取脚本文件的操作。另外，脚本文件在预先设置完成后，可放置于文件服务器PackageServer中，PackageServer为高速下载服务器，可用于放置各类二进制文件及脚本文件，其内的内容通常固定不变，故可应用在本申请实施例中，提升了脚本文件获取的实时性和可靠性。在此基础上，由于Kubernetes平台适用于不同的操作系统，如Windows、Linux等，而不同的环境下，底层的搭建代码存在一定程度的差异，故可针对多个操作系统编写与其对应的多个脚本文件，并将多个脚本文件放置于PackageServer中。在接收到用户通过终端设备发出的关于Kubernetes主节点的搭建指令后，识别终端设备所属操作系统，并获取PackageServer上存储的与终端设备的操作系统对应的脚本文件，防止获取到的脚本文件与运行环境不匹配，无法执行。

[0029] S102：执行所述脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数，生成配置文件。

[0030] 在获取到预设的脚本文件后，执行该脚本文件，在该脚本文件的执行过程中获取与Kubernetes主节点相关的创建参数，部分创建参数如Kubernetes主节点的版本号等可自动获取，另一部分创建参数如日志文件路径等需要检测用户输入，并从用户的输入内容中查找与日志文件路径相关的内容。其中，与Kubernetes主节点相关的创建参数包括Kubernetes主节点的属性参数，比如安装版本、Kubernetes主节点的互联网协议地址、运行Kubernetes主节点的主机标识（名称）以及区域标识等，还包括构成Kubernetes主节点的各个组件的属性参数等。当获取到用户输入的创建参数后，脚本文件根据该创建参数生成配置文件。配置文件实质利用脚本语言编写而成，与脚本文件相似，为了便于说明，可将配置文件命名为部署脚本，在后续步骤中，配置文件执行，以通过SaltStack工具搭建Kubern

etes主节点。

[0031] 可选地，根据与Kubernetes主节点相关的创建参数的各个类型生成输入模板，并获取用户通过输入模板输入的创建参数。由于创建参数的各个类型确定，故可根据创建参数的类型生成输入模板，输入模板中包含创建参数的各个类型的名称，便于用户输入与各个类型的名称对应的内容，降低了创建参数输入出错的可能。此外，输入模板还可配置于脚本文件中，当需要获取创建参数时，脚本文件执行，自动提供输入模板至用户。

[0032] S103：启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点。

[0033] 以往的如kops等工具虽然能够进行对Kubernetes主节点的搭建，但是在具体的搭建过程中，需要用户编写代码完成对如Kubernetes主节点互联网协议地址的配置。故在本申请实施例中，使用SaltStack工具搭建Kubernetes主节点。需要说明的是，SaltStack工具为开源的基础设施管理工具，分为管理主节点（SaltMaster）和管理从节点（SaltMinion）两个部分。与Kubernetes主节点和Kubernetes从节点的关系类似，SaltStack工具的管理主节点主要对管理从节点进行统一管理，而管理从节点运行于需要管理的物理服务器或虚拟机上，通过执行管理主节点传输的指令，并根据配置文件搭建Kubernetes主节点，具体过程在后文进行阐述。

[0034] 通过图1所示实施例可知，在本申请实施例中，通过获取预设的用于搭建Kubernetes主节点的脚本文件，脚本文件可存储于文件服务器PackageServer中，并获取与Kubernetes主节点相关的创建参数，在执行脚本文件后，生成与创建参数对应的配置文件，最后启动SaltStack工具，具体通过SaltStack工具内的管理主节点和管理从节点，根据配置文件搭建Kubernetes主节点，实现了Kubernetes主节点的自动化搭建，减少了用户操作，提升了搭建效率。

[0035] 请参阅图2，图2是本申请实施例二提供的一种自动搭建Kubernetes主节点的方法的实现流程图。相对于图1对应的实施例，本实施例对S103进行细化后得到S201~S202，详述如下：

[0036] S201：在所述SaltStack工具下配置管理主节点和多个管理从节点，其中，所述管理主节点与所述多个管理从节点连接。

[0037] SaltStack工具应用客户端/服务器架构，可实现远程命令执行和配置管理等功能

。SaltStack工具下包含管理主节点和多个管理从节点，管理主节点相当于SaltStack工具的服务器，负责资源管理，多个管理从节点相当于SaltStack工具的客户端，负责执行管理主节点下发的指令。在本其实施例中，将为用户提供具体服务的区域称为可用区，并将可用区内部具体分为资源管理区和公共服务区，其中，资源管理区用于部署资源管理相关的程序与组件，可以访问公共服务区的物理服务器或虚拟机，相当于客户端/服务器架构中的服务器，故管理主节点运行于资源管理区；公共服务区用于部署为用户提供具体服务的程序或服务，相当于客户端/服务器架构中的客户端，故在公共服务区安装管理从节点。在SaltStack工具启动后，自动配置管理主节点和多个管理从节点，其中，管理从节点在启动时，会根据加密算法生成私钥和公钥，生成完毕后并将公钥发送至管理主节点，管理主节点验证并接受公钥，从而与管理从节点建立连接。值得一提的是，管理主节点与管理从节点之间通过消息队列ZeroMQ传递消息，也即管理主节点将消息发布至消息队列ZeroMQ，管理从节点通过订阅消息队列ZeroMQ从而得到该消息。

[0038] S202：调用所述管理主节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点。

[0039] SaltStack工具下的管理主节点和多个管理从节点配置完成后，SaltStack工具调用管理主节点，管理主节点从多个管理从节点中选择一个，被选择的管理从节点接收搭建的指令，根据配置文件搭建Kubernetes主节点，实质是在该管理从节点所在的物理服务器或虚拟机进行Kubernetes主节点的搭建。

[0040] 通过图2所示实施例可知，在本申请实施例中，通过在SaltStack工具下配置管理主节点和多个管理从节点，管理主节点和多个管理从节点相连接，通过消息队列ZeroMQ实现通信，并调用管理主节点，使管理主节点从多个管理从节点中选取一个，在选取的管理从节点下根据配置文件搭建Kubernetes主节点，体现了通过SaltStack工具搭建Kubernetes主节点的具体过程，提升了搭建Kubernetes主节点的稳定性。

[0041] 请参阅图3，图3是本申请实施例三提供的一种自动搭建Kubernetes主节点的方法的实现流程图。相对于图2对应的实施例，本实施例对S202进行细化后得到S3

01~S303, 详述如下:

[0042] S301: 检测所述多个管理从节点下是否存在所述Kubernetes主节点。

[0043] 由于创建参数与Kubernetes主节点相关, 故在启动SaltStack工具, 并配置完成管理主节点和多个管理从节点后, 由管理主节点在与其连接的多个管理从节点下查找是否存在与创建参数对应的Kubernetes主节点。

[0044] S302: 若所述多个管理从节点下都不存在所述Kubernetes主节点, 则从所述多个管理从节点中选取与所述创建参数对应的管理从节点, 作为搭建从节点。

[0045] 如果在所有的管理从节点下都未查找到Kubernetes主节点, 则证明Kubernetes主节点未被创建。值得一提的是, 由于在本申请实施例中, 是在管理从节点下搭建Kubernetes主节点, 故在步骤S102中, 获取的与Kubernetes主节点相关的创建参数也包含用户配置的管理从节点相关的参数, 那么在本步骤中, 即可根据创建参数从多个管理从节点中选择相应的管理从节点, 作为搭建从节点。若检测到创建参数不包含与管理从节点相关的参数, 则可从多个管理从节点中随机选择其中的一个作为搭建从节点, 也可根据预设的选择机制在多个管理从节点中进行选择。

[0046] S303: 在所述搭建从节点下根据所述配置文件搭建所述Kubernetes主节点。

[0047] 选取搭建从节点后, 在该搭建从节点下根据配置文件搭建Kubernetes主节点。此外, 可对该搭建从节点设置标记, 具体可对该搭建从节点设置标记戳, 证明该搭建从节点已被用于创建Kubernetes主节点。在标记戳设置完成后, 若再次设置新的Kubernetes主节点, 那么在执行步骤S301的操作, 即在检测多个管理从节点下是否存在Kubernetes主节点时, 则可根据管理从节点是否存在标记戳来判断该管理从节点是否已用于创建Kubernetes主节点, 提升了检测效率。

[0048] 通过图3所示实施例可知, 在本申请实施例中, 通过检测在多个管理从节点下是否存在Kubernetes主节点, 防止重复搭建造成的资源浪费。

[0049] 请参阅图4, 图4是本申请实施例四提供的一种自动搭建Kubernetes主节点的方法的实现流程图。相对于图1对应的实施例, 本实施例在Kubernetes主节点包括数据库组件、接口服务组件、控制组件以及调度组件的基础上, 对S103进行细化后得到S401~S402, 详述如下:

- [0050] S401: 通过所述SaltStack工具部署所述数据库组件, 并获取与所述创建参数相关的网络配置数据, 将所述网络配置数据加载至所述数据库组件。
- [0051] 在本申请实施例中, Kubernetes主节点包括数据库(Etcd)组件、接口服务(Kube ApiServer)组件、控制(Kube Controller Manager)组件以及调度(Kube Scheduler)组件, 为了便于解释本申请实施例的内容, 提供了如图7所示的一个Kubernetes集群的示意图, 在图7中, Master代表Kubernetes主节点, Controller Manager代表控制组件, Scheduler代表调度组件, ApiServer代表接口服务组件, Etcd代表数据库组件, 最下方的Node代表Kubernetes从节点。值得一提的是, 在搭建Kubernetes主节点, 配置Kubernetes主节点下的各个组件前, 关闭防火墙, 基于防火墙策略实现最小化的网络打通, 即将管理主节点所在的资源管理区和管理从节点所在的公共服务区之间的网络打通。
- [0052] 数据库组件即Etcd组件是分布式键值存储服务, 通过分布式Raft算法来保证数据一致性, 并且提供基于超文本传输协议(HyperText Transfer Protocol, HTTP)和JavaScript对象标记语言(JavaScript Object Notation, JSON)的应用程序编程接口(Application Programming Interface, API), 在本申请实施例中, 数据库组件用于配置管理、数据存储及作为分布式锁, 可通过SaltStack工具进行部署, 具体地, 管理从节点根据配置文件配置数据库组件, 具体通过配置文件从文件服务器中获取数据库组件配置文件, 该数据库组件配置文件为二进制文件, 用于配置数据库组件服务, 配置内容还包括数据库名称和数据路径等。当数据库组件配置完成后, 获取与创建参数相关的网络配置数据, 并将网络配置数据加载至数据库组件。其中, 网络配置数据的获取方式可以是在脚本文件中预设网络配置模板, 并在获取到创建参数时, 提取出创建参数中与网络配置数据相关的内容, 并添加至网络配置模板中, 从而在需要网络配置数据时直接从网络配置模板中获取。在添加Kubernetes从节点, 待加入的节点向数据库组件发送请求时, 数据库组件可根据网络配置数据将待加入的节点识别为Kubernetes从节点。
- [0053] S402: 依次部署所述接口服务组件、所述控制组件以及所述调度组件, 并修改所述接口服务组件的服务地址。
- [0054] 数据库组件部署完成后, 继续部署接口服务组件。如图7所示, 在Kubernetes主

节点中，接口服务组件负责与数据库组件进行数据交互，值得一提的是，除了接口服务组件外，Kubernetes主节点的其他组件不会直接操作数据库组件。接口服务组件是Kubernetes主节点的数据中心，管理Kubernetes集群的应用程序编程接口。在步骤S102中提到，配置文件实质上是脚本语言编写的脚本文件，故在本步骤中，根据配置文件自动提取文件服务器中的接口服务配置文件，并根据接口服务配置文件对接口服务组件进行配置。另外，用户输入的创建参数包含数据库组件地址和接口服务地址，而配置文件由创建参数生成，故在步骤中还通过配置文件在接口服务组件中配置数据库组件地址和接口服务组件地址，其中，数据库组件地址为数据库组件中提供数据库服务的地址，在接口服务组件配置该数据库组件地址，以使接口服务组件能够通过数据库组件地址访问数据库组件；接口服务组件地址为接口服务组件的本地地址。值得一提的是，接口服务组件包括两类通信接口，一类是用于为Kubernetes从节点提供访问数据库组件的接口，需要通过安全证书进行权限认证，具体内容在后文进行介绍；另一类是用于与调度组件和控制组件进行数据交互的接口，此接口为组件内部接口，不需进行权限认证。

[0055] 控制组件为Kubernetes集群的管理控制中心，保证Kubernetes集群中各种资源的状态处于正常状态，当监控到Kubernetes集群中某个资源状态不正常时，控制组件会触发调度操作，控制组件包括节点控制器（Node Controller）和副本控制器（Replication Controller）等。调度组件负责对Kubernetes集群中的容器进行编排，具体负责将容器调度到具体的Kubernetes从节点。调度组件通过接口服务组件提供的应用程序编程接口监听容器，获取待调度容器，并根据预设的排序机制对各个Kubernetes从节点进行排序，将容器调度至排序首位的Kubernetes从节点。通过配置文件从文件服务器中获取控制组件配置文件和调度组件配置文件，控制组件配置文件和调度组件配置文件都为二进制文件，其中控制组件配置文件用于对控制组件相关服务进行配置，调度组件配置文件用于对调度组件相关进行配置。配置过程还包括设置环境变量，环境变量包括控制组件和调度组件的版本环境变量等，环境变量由用户输入的创建参数确定，并通过配置文件在控制组件和调度组件中进行环境变量的自动配置。此外，根据配置文件自动修

改接口服务组件的服务地址，以便控制组件和调度组件分别能够与接口服务组件进行数据交互，其中服务地址是指接口服务组件向外提供服务的地址。在配置完成Kubernetes主节点下的所有组件后，可创建系统域名系统（Domain Name System, DNS）服务，也即kube-dns，负责在Kubernetes集群内部创建DNS服务，使得容器之间通过域名查找服务，便于容器之间构建服务。可选地，创建dashboard，dashboard是Kubernetes集群的用户界面，便于用户通过dashboard对Kubernetes集群的资源进行查看及操作。

[0056] 通过图4所示实施例可知，在本申请实施例中，Kubernetes主节点包括数据库组件、接口服务组件、控制组件和调度组件，通过SaltStack工具部署数据库组件，并从网络配置模板中获取与创建参数相关的网络配置数据，将网络配置数据加载至所述数据库组件，数据库组件配置完成后，依次部署接口服务组件、控制组件和调度组件，并修改接口服务组件的服务地址，便于控制组件和调度组件单独和接口服务组件进行数据交互，体现了Kubernetes主节点下各个组件的部署过程，提升了组件配置的自动化程度。

[0057] 请参阅图5，图5是本申请实施例五提供的一种自动搭建Kubernetes主节点的方法的实现流程图。相对于图4对应的实施例，本实施例对S401进行细化后得到S501~S502，详述如下：

[0058] S501：根据所述配置文件生成第一安全证书和第二安全证书，其中，第一安全证书用于控制对所述数据库组件的访问权限，第二安全证书用于控制对所述接口服务组件的访问权限。

[0059] 由于数据库组件和接口服务组件经常进行数据交互，故为了提升上述两个组件的安全性，在部署数据库组件时，根据配置文件自动生成第一安全证书和第二安全证书，第一安全证书和第二安全证书都为安全套接字层（Secure Socket Layer, SSL）证书。其中，第一安全证书用于控制对数据库组件的访问权限，第二安全证书用于控制对接口服务组件的访问权限。

[0060] S502：将所述第一安全证书配置至所述数据库组件和所述接口服务组件，并将所述第二安全证书配置至所述接口服务组件。

[0061] 将第一安全证书配置至数据库组件和接口服务组件中，配置完成并启动数据库

组件服务后，Kubernetes从节点和接口服务组件在与数据库组件进行数据交互之前，都需要经过第一安全证书的验证。将第二安全证书配置至接口服务组件中，在启动接口服务组件服务后，Kubernetes从节点在与接口服务组件进行数据交互之前，都需要经过第二安全证书的验证。

[0062] 通过图5所示实施例可知，在本申请实施例中，在部署数据库组件时，生成第一安全证书和第二安全证书，并将第一安全证书配置至数据库组件和接口服务组件，用于控制Kubernetes从节点和接口服务组件对数据库组件的访问权限，将第二安全证书配置至接口服务组件，用于控制Kubernetes从节点对接口服务组件的访问权限，提升了数据库组件和接口服务组件的访问安全性。

[0063] 在本申请另一实施例中，可基于专有网络实现Kubernetes主节点及Kubernetes集群的搭建，如图8所示，提供了基于Kubernetes集群的区域结构图。首先对图8中的各个结构进行说明，用户专有网络（Virtual Private Cloud，VPC）是用户创建的自定义私有网络，不同的专有网络之间逻辑隔离，用户可在自己的专有网络中创建和管理云主机，实现负载均衡、安全访问控制等功能，VPC1和VPC2是不同用户创建的专有网络。可用区包括资源管理区和公共服务区，以及具体的用户VPC区域，为云平台服务提供具体服务的区域，图8中的ECA and SCA为可用区的两个实例，分别代表华东区A和华南区A。如前所述，资源管理区用于部署资源管理相关的程序与组件，可以访问公共服务区的物理服务器或虚拟机，而公共服务区用于部署为用户提供具体服务的程序或服务，可以与用户VPC互通，公共服务区内包含资源池，资源池为可运行的物理服务器或虚拟机的集合，Package Server代表文件服务器，资源池内的master1和master2等代表Kubernetes主节点，而VPC1和VPC2中的node代表Kubernetes从节点，不同用户的用户VPC之间默认隔离。管理区是相对于可用区更高一层的统称，能够对所有可用区的资源管理区和公共服务区打通并进行管理，图8中管理区内的Manager代表Kubernetes Manager，是能够管理Kubernetes集群的管理程序。

[0064] 另外，图8的管理区中与Manager连接的DB代表数据库，与Manager连接的etcd集群代表数据库组件集群。其中，数据库用于存储数据，数据库组件集群作为分布式锁保证一致性。Kubernetes Manager通过调用资源管理区的管理主节点来

对公共服务区的物理服务器或虚拟机进行管理，公共服务区的资源池内存在多个管理从节点，当Kubernetes Manager部署Kubernetes主节点时，从资源池的多个管理从节点中选取可用的管理从节点，并在该管理从节点部署Kubernetes主节点。部署完成后，用户VPC下的节点作为Kubernetes从节点，与Kubernetes主节点组成Kubernetes集群。本申请实施例通过在用户的专有网络下搭建Kubernetes主节点以及Kubernetes集群，保证了不同Kubernetes集群间的隔离性。

[0065] 对应于上文实施例所述的一种自动搭建Kubernetes主节点的方法，图6示出了本申请实施例提供的一种自动搭建Kubernetes主节点的装置的一个结构框图，参照图6，该装置包括：

[0066] 第一获取单元61，用于获取预设的用于搭建Kubernetes主节点的脚本文件；

[0067] 第二获取单元62，用于执行所述脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数，生成配置文件；

[0068] 搭建单元63，用于启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点。

[0069] 可选地，搭建单元63包括：

[0070] 节点配置单元，用于在所述SaltStack工具下配置管理主节点和多个管理从节点，其中，所述管理主节点与所述多个管理从节点连接；

[0071] 调用单元，用于调用所述管理主节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点。

[0072] 可选地，调用单元包括：

[0073] 检测单元，用于检测所述多个管理从节点下是否存在所述Kubernetes主节点；

[0074] 选取单元，用于若所述多个管理从节点下都不存在所述Kubernetes主节点，则从所述多个管理从节点中选取与所述创建参数对应的管理从节点，作为搭建从节点；

[0075] 搭建子单元，用于在所述搭建从节点下根据所述配置文件搭建所述Kubernetes主节点。

[0076] 可选地，Kubernetes主节点包括数据库组件、接口服务组件、控制组件以及调度组件，搭建单元63包括：

- [0077] 部署单元，用于通过所述SaltStack工具部署所述数据库组件，并获取与所述创建参数相关的网络配置数据，将所述网络配置数据加载至所述数据库组件；
- [0078] 依次部署单元，用于依次部署所述接口服务组件、所述控制组件以及所述调度组件，并修改所述接口服务组件的服务地址。
- [0079] 可选地，部署单元还包括：
- [0080] 生成单元，用于根据所述配置文件生成第一安全证书和第二安全证书，其中，第一安全证书用于控制对所述数据库组件的访问权限，第二安全证书用于控制对所述接口服务组件的访问权限；
- [0081] 配置单元，用于将所述第一安全证书配置至所述数据库组件和所述接口服务组件，并将所述第二安全证书配置至所述接口服务组件。
- [0082] 图9是本申请实施例提供的终端设备的示意图。如图9所示，该实施例的终端设备9包括：处理器90以及存储器91，所述存储器91中存储有可在所述处理器90上运行的计算机可读指令92，例如搭建Kubernetes主节点的程序。所述处理器90执行所述计算机可读指令92时实现上述各个自动搭建Kubernetes主节点的方法实施例中的步骤，例如图1所示的步骤S101至S103。或者，所述处理器90执行所述计算机可读指令92时实现上述装置实施例中各单元的功能，例如图6所示单元61至63的功能。
- [0083] 示例性的，所述计算机可读指令92可以被分割成一个或多个模块/单元，所述一个或者多个模块/单元被存储在所述存储器91中，并由所述处理器90执行，以完成本申请。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机可读指令段，该指令段用于描述所述计算机可读指令92在所述终端设备9中的执行过程。例如，所述计算机可读指令92可以被分割成第一获取单元、第二获取单元及搭建单元，各单元具体功能如上所述。
- [0084] 所述终端设备可包括，但不限于，处理器90、存储器91。本领域技术人员可以理解，图9仅仅是终端设备9的示例，并不构成对终端设备9的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件，例如所述终端设备还可以包括输入输出设备、网络接入设备、总线等。
- [0085] 所称处理器90可以是中央处理单元(Central Processing Unit, CPU)，还可以是其

他通用处理器、数字信号处理器 (Digital Signal Processor, DSP)、专用集成电路 (Application Specific Integrated Circuit, ASIC)、现成可编程门阵列 (Field-Programmable Gate Array, FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0086] 所述存储器91可以是所述终端设备9的内部存储单元，例如终端设备9的硬盘或内存。所述存储器91也可以是所述终端设备9的外部存储设备，例如所述终端设备9上配备的插接式硬盘，智能存储卡 (Smart Media Card, SMC)，安全数字 (Secure Digital, SD) 卡，闪存卡 (Flash Card) 等。进一步地，所述存储器91还可以既包括所述终端设备9的内部存储单元也包括外部存储设备。所述存储器91用于存储所述计算机可读指令以及所述终端设备所需的其他程序和数据。所述存储器91还可以用于暂时地存储已经输出或者将要输出的数据。

[0087] 另外，在本申请各个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。

[0088] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备 (可以是个人计算机，服务器，或者网络设备) 执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U盘、移动硬盘、只读存储器 (Read-Only Memory, ROM)、随机存取存储器 (Random Access Memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0089] 以上所述，以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术

特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

权利要求书

- [权利要求 1] 一种自动搭建Kubernetes主节点的方法，其特征在于，包括：
获取预设的用于搭建Kubernetes主节点的脚本文件；
执行所述脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数，生成配置文件；
启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点。
- [权利要求 2] 如权利要求1所述的方法，其特征在于，所述启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点，包括：
在所述SaltStack工具下配置管理主节点和多个管理从节点，其中，所述管理主节点与所述多个管理从节点连接；
调用所述管理主节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点。
- [权利要求 3] 如权利要求2所述的方法，其特征在于，所述调用所述管理主节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点，包括：
检测所述多个管理从节点下是否存在所述Kubernetes主节点；
若所述多个管理从节点下都不存在所述Kubernetes主节点，则从所述多个管理从节点中选取与所述创建参数对应的管理从节点，作为搭建从节点；
在所述搭建从节点下根据所述配置文件搭建所述Kubernetes主节点。
- [权利要求 4] 如权利要求1所述的方法，所述Kubernetes主节点包括数据库组件、接口服务组件、控制组件以及调度组件，其特征在于，所述启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点，包括：
通过所述SaltStack工具部署所述数据库组件，并获取与所述创建参数相关的网络配置数据，将所述网络配置数据加载至所述数据库组件；
依次部署所述接口服务组件、所述控制组件以及所述调度组件，并修改所述接口服务组件的服务地址。

- [权利要求 5] 如权利要求4所述的方法，其特征在于，所述通过所述SaltStack工具部署所述数据库组件，还包括：
根据所述配置文件生成第一安全证书和第二安全证书，其中，第一安全证书用于控制对所述数据库组件的访问权限，第二安全证书用于控制对所述接口服务组件的访问权限；
将所述第一安全证书配置至所述数据库组件和所述接口服务组件，并将所述第二安全证书配置至所述接口服务组件。
- [权利要求 6] 一种自动搭建Kubernetes主节点的装置，其特征在于，包括：
第一获取单元，用于获取预设的用于搭建Kubernetes主节点的脚本文件；
第二获取单元，用于执行所述脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数，生成配置文件；
搭建单元，用于启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点。
- [权利要求 7] 如权利要求6所述的装置，其特征在于，所述搭建单元，还包括：
节点配置单元，用于在所述SaltStack工具下配置管理主节点和多个管理从节点，其中，所述管理主节点与所述多个管理从节点连接；
调用单元，用于调用所述管理主节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点。
- [权利要求 8] 如权利要求7所述的装置，其特征在于，所述调用单元，包括：
检测单元，用于检测所述多个管理从节点下是否存在所述Kubernetes主节点；
选取单元，用于若所述多个管理从节点下都不存在所述Kubernetes主节点，则从所述多个管理从节点中选取与所述创建参数对应的管理从节点，作为搭建从节点；
搭建子单元，用于在所述搭建从节点下根据所述配置文件搭建所述Kubernetes主节点。
- [权利要求 9] 如权利要求6所述的装置，所述Kubernetes主节点包括数据库组件、接

口服务组件、控制组件以及调度组件，其特征在于，所述搭建单元，包括：

部署单元，用于通过所述SaltStack工具部署所述数据库组件，并获取与所述创建参数相关的网络配置数据，将所述网络配置数据加载至所述数据库组件；

依次部署单元，用于依次部署所述接口服务组件、所述控制组件以及所述调度组件，并修改所述接口服务组件的服务地址。

[权利要求 10]

如权利要求9所述的装置，其特征在于，所述部署单元，还包括：

生成单元，用于根据所述配置文件生成第一安全证书和第二安全证书，其中，第一安全证书用于控制对所述数据库组件的访问权限，第二安全证书用于控制对所述接口服务组件的访问权限；

配置单元，用于将所述第一安全证书配置至所述数据库组件和所述接口服务组件，并将所述第二安全证书配置至所述接口服务组件。

[权利要求 11]

一种终端设备，其特征在于，包括存储器以及处理器，所述存储器中存储有可在所述处理器上运行的计算机可读指令，所述处理器执行所述计算机可读指令时实现如下步骤：

获取预设的用于搭建Kubernetes主节点的脚本文件；

执行所述脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数，生成配置文件；

启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点

。

[权利要求 12]

根据权利要求11所述的终端设备，其特征在于，所述启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点，包括：

在所述SaltStack工具下配置管理主节点和多个管理从节点，其中，所述管理主节点与所述多个管理从节点连接；

调用所述管理主节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点。

[权利要求 13]

根据权利要求12所述的终端设备，其特征在于，所述调用所述管理主

节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点，包括：

检测所述多个管理从节点下是否存在所述Kubernetes主节点；

若所述多个管理从节点下都不存在所述Kubernetes主节点，则从所述多个管理从节点中选取与所述创建参数对应的管理从节点，作为搭建从节点；

在所述搭建从节点下根据所述配置文件搭建所述Kubernetes主节点。

[权利要求 14]

根据权利要求11所述的终端设备，所述Kubernetes主节点包括数据库组件、接口服务组件、控制组件以及调度组件，其特征在于，所述启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点，包括：

通过所述SaltStack工具部署所述数据库组件，并获取与所述创建参数相关的网络配置数据，将所述网络配置数据加载至所述数据库组件；依次部署所述接口服务组件、所述控制组件以及所述调度组件，并修改所述接口服务组件的服务地址。

[权利要求 15]

根据权利要求14所述的终端设备，其特征在于，所述通过所述SaltStack工具部署所述数据库组件，还包括：

根据所述配置文件生成第一安全证书和第二安全证书，其中，第一安全证书用于控制对所述数据库组件的访问权限，第二安全证书用于控制对所述接口服务组件的访问权限；

将所述第一安全证书配置至所述数据库组件和所述接口服务组件，并将所述第二安全证书配置至所述接口服务组件。

[权利要求 16]

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可读指令，其特征在于，所述计算机可读指令被至少一个处理器执行时实现如下步骤：

获取预设的用于搭建Kubernetes主节点的脚本文件；

执行所述脚本文件，以根据用户输入的与所述Kubernetes主节点相关的创建参数，生成配置文件；

启动SaltStack工具，以根据所述配置文件搭建所述Kubernetes主节点

。

[权利要求 17] 根据权利要求16所述的计算机可读存储介质，其特征在于，所述计算机可读指令被至少一个处理器执行时实现如下步骤：

在所述SaltStack工具下配置管理主节点和多个管理从节点，其中，所述管理主节点与所述多个管理从节点连接；

调用所述管理主节点，以使所述多个管理从节点中的一个根据所述配置文件搭建所述Kubernetes主节点。

[权利要求 18] 根据权利要求17所述的计算机可读存储介质，其特征在于，所述计算机可读指令被至少一个处理器执行时实现如下步骤：

检测所述多个管理从节点下是否存在所述Kubernetes主节点；

若所述多个管理从节点下都不存在所述Kubernetes主节点，则从所述多个管理从节点中选取与所述创建参数对应的管理从节点，作为搭建从节点；

在所述搭建从节点下根据所述配置文件搭建所述Kubernetes主节点。

[权利要求 19] 根据权利要求16所述的计算机可读存储介质，所述Kubernetes主节点包括数据库组件、接口服务组件、控制组件以及调度组件，其特征在于，所述计算机可读指令被至少一个处理器执行时实现如下步骤：

通过所述SaltStack工具部署所述数据库组件，并获取与所述创建参数相关的网络配置数据，将所述网络配置数据加载至所述数据库组件；

依次部署所述接口服务组件、所述控制组件以及所述调度组件，并修改所述接口服务组件的服务地址。

[权利要求 20] 根据权利要求19所述的计算机可读存储介质，其特征在于，所述计算机可读指令被至少一个处理器执行时还实现如下步骤：

根据所述配置文件生成第一安全证书和第二安全证书，其中，第一安全证书用于控制对所述数据库组件的访问权限，第二安全证书用于控制对所述接口服务组件的访问权限；

将所述第一安全证书配置至所述数据库组件和所述接口服务组件，并

将所述第二安全证书配置至所述接口服务组件。

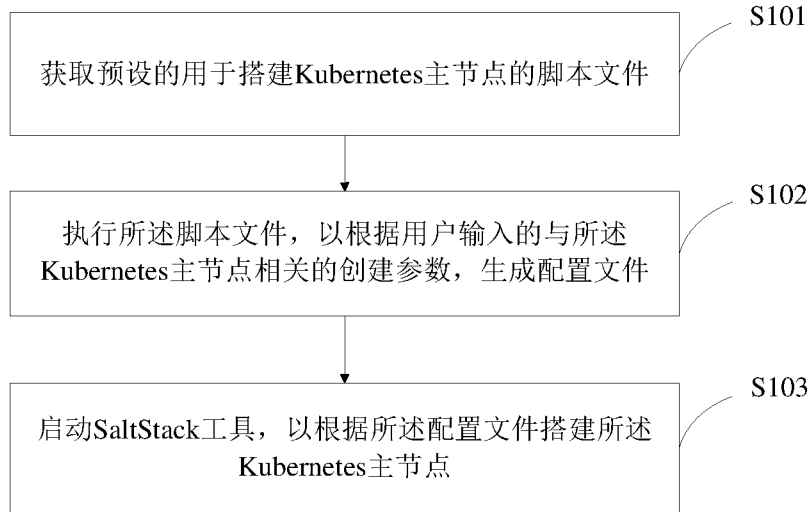


图 1

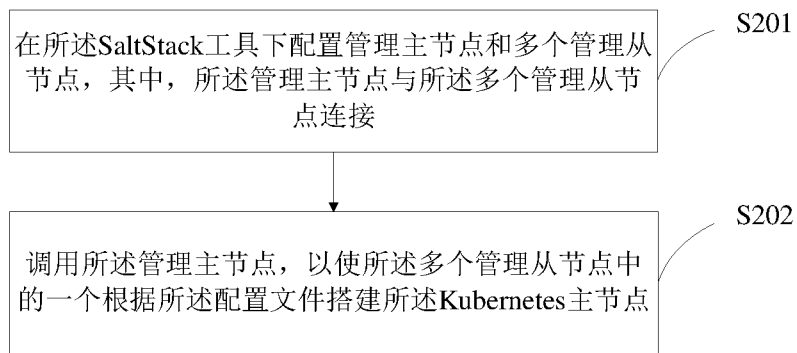


图 2

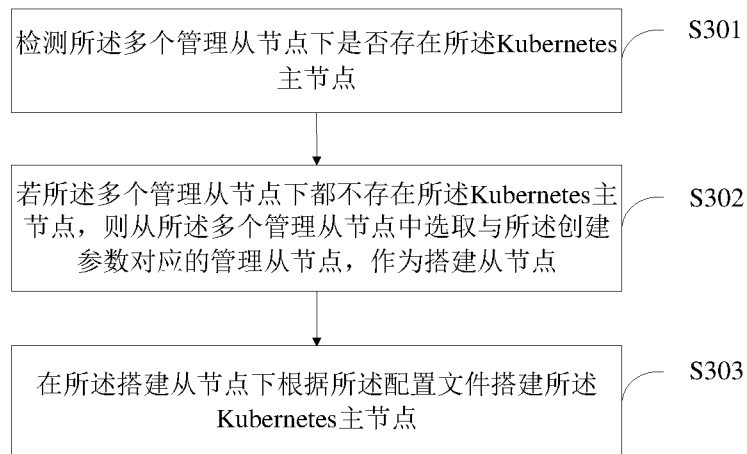


图 3

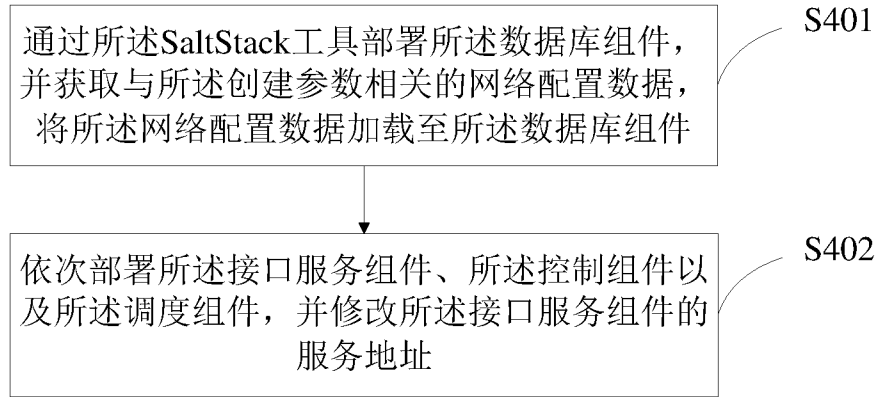


图 4

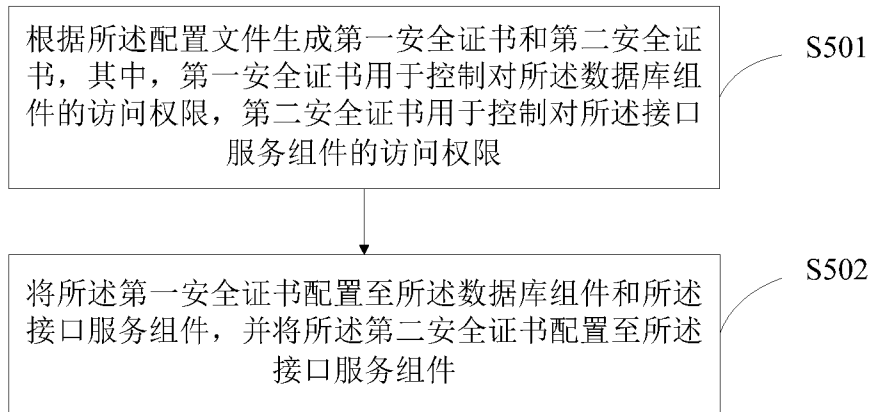


图 5

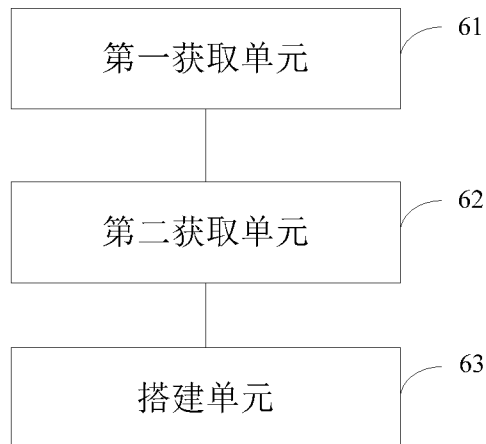


图 6

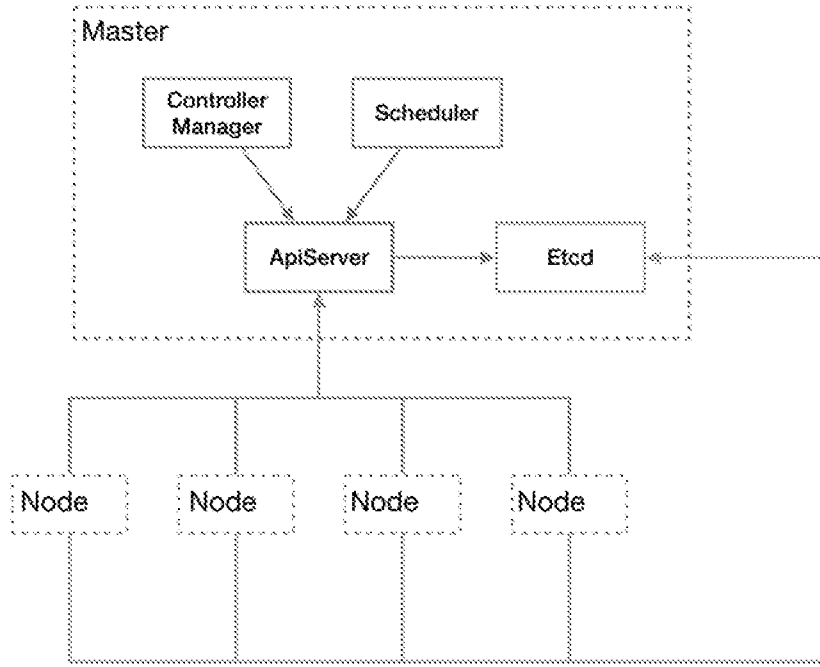


图 7

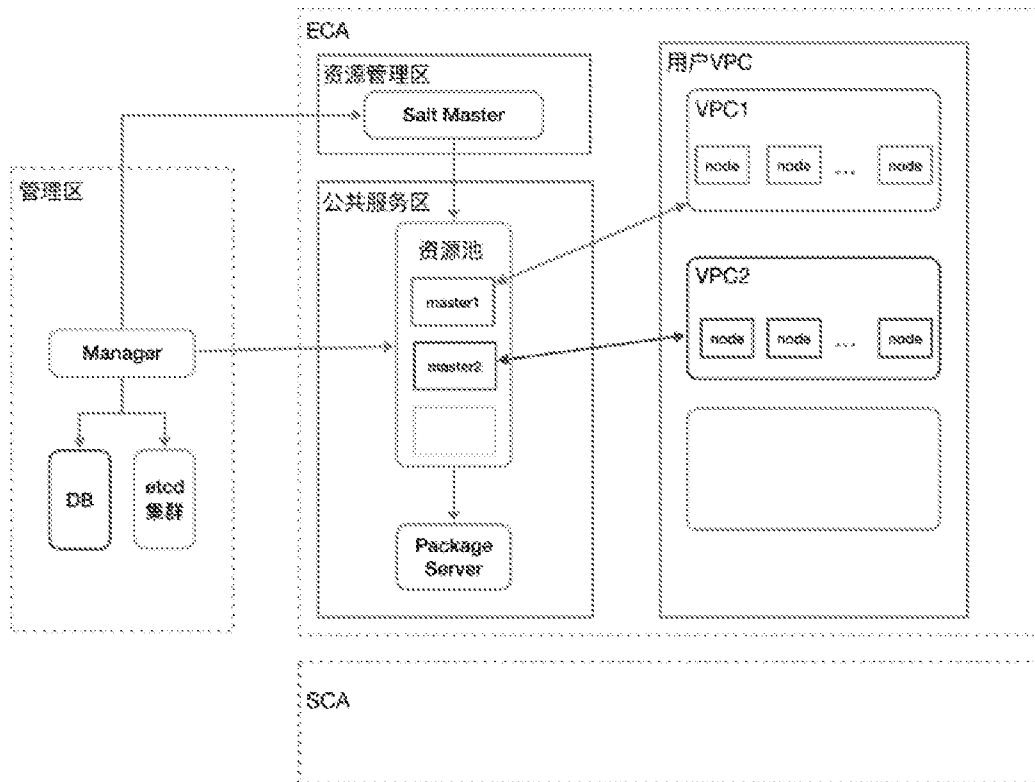


图 8

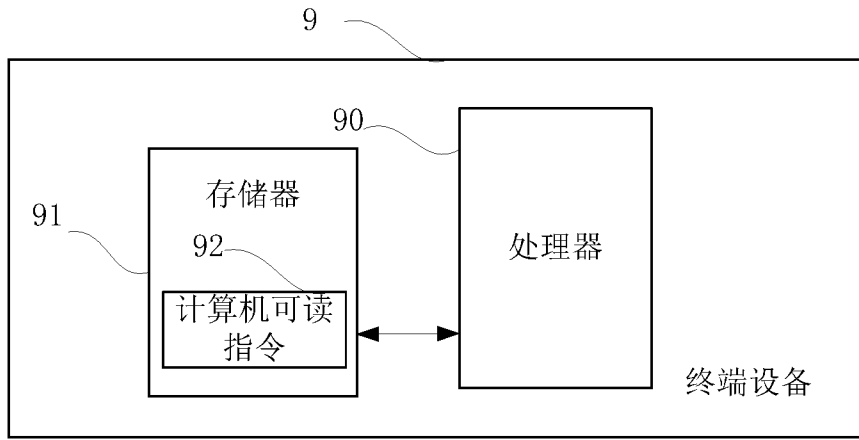


图 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/093710

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/24(2006.01)i; G06F 9/455(2006.01)i

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)

H04L; 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)

CNABS; CNTXT; CNKI; VEN; WOTXT; EPTXT; USTXT; IEEE: Kubernetes, 主, 从, 脚本, 参数, SaltStack, master, slave, script, shell, parameter

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 106850621 A (NANJING INNOVATIVE DATA TECHNOLOGIES INC.) 13 June 2017 (2017-06-13) description, paragraphs [0019]-[0027]	1, 2, 4-7, 9-12, 14-17, 19, 20
Y	CN 105553737 A (INSPUR (BEIJING) ELECTRONIC INFORMATION INDUSTRY CO., LTD.) 04 May 2016 (2016-05-04) description, paragraphs [0027]-[0040]	1, 2, 4-7, 9-12, 14-17, 19, 20
A	US 2017111241 A1 (DRAIOS INC.) 20 April 2017 (2017-04-20) entire document	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

10 December 2018

Date of mailing of the international search report

19 December 2018

Name and mailing address of the ISA/CN

**National Intellectual Property Administration, PRC (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088
China**

Authorized officer

Facsimile No. (86-10)62019451

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/093710

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN	106850621	A	13 June 2017	None	
CN	105553737	A	04 May 2016	None	
US	2017111241	A1	20 April 2017	None	

国际检索报告

国际申请号

PCT/CN2018/093710

<p>A. 主题的分类 H04L 12/24(2006.01)i; G06F 9/455(2006.01)i</p> <p>按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类</p>														
<p>B. 检索领域</p> <p>检索的最低限度文献(标明分类系统和分类号) H04L; G06F</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用)) CNABS;CNTXT;CNKI;VEN;WOTXT;EPTXT;USTXT;IEEE: Kubernetes, 主, 从, 脚本, 参数, SaltStack, master, slave, script, shell, parameter</p>														
<p>C. 相关文件</p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>Y</td> <td>CN 106850621 A (南京云创大数据科技股份有限公司) 2017年 6月 13日 (2017 - 06 - 13) 说明书19-27段</td> <td>1、2、4-7、9-12、14-17、19、20</td> </tr> <tr> <td>Y</td> <td>CN 105553737 A (浪潮北京电子信息产业有限公司) 2016年 5月 4日 (2016 - 05 - 04) 说明书27-40段</td> <td>1、2、4-7、9-12、14-17、19、20</td> </tr> <tr> <td>A</td> <td>US 2017111241 A1 (DRAIOS INC.) 2017年 4月 20日 (2017 - 04 - 20) 全文</td> <td>1-20</td> </tr> </tbody> </table>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	Y	CN 106850621 A (南京云创大数据科技股份有限公司) 2017年 6月 13日 (2017 - 06 - 13) 说明书19-27段	1、2、4-7、9-12、14-17、19、20	Y	CN 105553737 A (浪潮北京电子信息产业有限公司) 2016年 5月 4日 (2016 - 05 - 04) 说明书27-40段	1、2、4-7、9-12、14-17、19、20	A	US 2017111241 A1 (DRAIOS INC.) 2017年 4月 20日 (2017 - 04 - 20) 全文	1-20
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求												
Y	CN 106850621 A (南京云创大数据科技股份有限公司) 2017年 6月 13日 (2017 - 06 - 13) 说明书19-27段	1、2、4-7、9-12、14-17、19、20												
Y	CN 105553737 A (浪潮北京电子信息产业有限公司) 2016年 5月 4日 (2016 - 05 - 04) 说明书27-40段	1、2、4-7、9-12、14-17、19、20												
A	US 2017111241 A1 (DRAIOS INC.) 2017年 4月 20日 (2017 - 04 - 20) 全文	1-20												
<p><input type="checkbox"/> 其余文件在C栏的续页中列出。</p> <p><input checked="" type="checkbox"/> 见同族专利附件。</p>														
<p>* 引用文件的具体类型:</p> <p>“A” 认为不特别相关的表示了现有技术一般状态的文件</p> <p>“E” 在国际申请日的当天或之后公布的在先申请或专利</p> <p>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)</p> <p>“O” 涉及口头公开、使用、展览或其他方式公开的文件</p> <p>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件</p> <p>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件</p> <p>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性</p> <p>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性</p> <p>“&” 同族专利的文件</p>														
<p>国际检索实际完成的日期</p> <p>2018年 12月 10日</p>		<p>国际检索报告邮寄日期</p> <p>2018年 12月 19日</p>												
<p>ISA/CN的名称和邮寄地址</p> <p>中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088</p> <p>传真号 (86-10)62019451</p>		<p>受权官员</p> <p>李倩</p> <p>电话号码 86-(010)-62089374</p>												

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2018/093710

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	106850621	A	2017年 6月 13日	无	
CN	105553737	A	2016年 5月 4日	无	
US	2017111241	A1	2017年 4月 20日	无	