



(12)发明专利

(10)授权公告号 CN 106330923 B

(45)授权公告日 2019.10.25

(21)申请号 201610744129.2

(22)申请日 2016.08.26

(65)同一申请的已公布的文献号
申请公布号 CN 106330923 A

(43)申请公布日 2017.01.11

(73)专利权人 中国联合网络通信集团有限公司
地址 100033 北京市西城区金融大街21号

(72)发明人 李铭轩 魏进武 张呈宇 张基恒
博格利

(74)专利代理机构 北京同立钧成知识产权代理有限公司 11205
代理人 杨泽 刘芳

(51)Int.Cl.
H04L 29/06(2006.01)
H04L 29/08(2006.01)

(56)对比文件

CN 105656653 A,2016.06.08,
CN 105743995 A,2016.07.06,
CN 105897946 A,2016.08.24,
CN 104703211 A,2015.06.10,
张俊.Kubernetes初探.《https://
blog.csdn.net/zhangjun2915/article/
details/40598151》.2014,全文.

审查员 吴玉婕

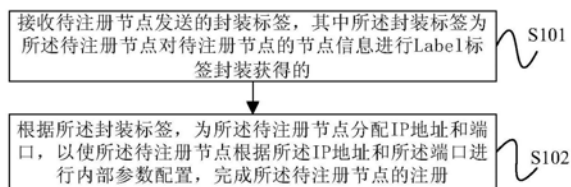
权利要求书1页 说明书4页 附图2页

(54)发明名称

基于Kubernetes系统的节点注册方法及系统

(57)摘要

本发明实施例提供一种基于Kubernetes系统的节点注册方法及系统,其中,该方法包括:接收待注册节点发送的封装标签,其中所述封装标签为所述待注册节点对所述待注册节点的节点信息进行Label标签封装获得的;根据所述封装标签,为所述待注册节点分配IP地址和端口,以使所述待注册节点根据所述IP地址和所述端口进行内部参数配置,完成所述待注册节点的注册。本发明实施例提供的基于Kubernetes系统的节点注册方法及系统,实现了Kubernetes系统下节点的自动注册,提高了Kubernetes系统下节点的注册效率。



1. 一种基于Kubernetes系统的节点注册方法,其特征在于,包括:

接收待注册节点发送的封装标签,其中所述封装标签为所述待注册节点对所述待注册节点的节点信息进行Label标签封装获得的;所述封装标签包括所述待注册节点的变量信息;所述变量信息包括变量的名称以及变量对应的数据参数;

根据所述封装标签,为所述待注册节点分配IP地址和端口,以使所述待注册节点根据所述IP地址和所述端口进行内部参数配置,完成所述待注册节点的注册;

所述接收待注册节点的封装标签之后,所述方法还包括:

通过第一代理服务器将所述封装标签发送至第二代理服务器,以使所述第二代理服务器根据所述变量信息对所述待注册节点进行管理。

2. 根据权利要求1所述的方法,其特征在于,所述封装标签包括所述待注册节点的临时IP地址和临时端口信息,其中,所述临时IP地址和临时端口为所述待注册节点接入系统时,所述待注册节点上的第三代理服务器为所述待注册节点临时分配的。

3. 一种Kubernetes系统,其特征在于,包括:

接收单元,用于接收待注册节点发送的封装标签,其中所述封装标签为所述待注册节点对所述待注册节点的节点信息进行Label标签封装获得的;所述封装标签包括所述待注册节点的变量信息;所述变量信息包括变量的名称以及变量对应的数据参数;

配置单元,用于根据所述封装标签,为所述待注册节点分配IP地址和端口,以使所述待注册节点根据所述IP地址和所述端口进行内部参数配置,完成所述待注册节点的注册;

所述系统还包括:

管理单元,其中所述管理单元包括第一代理服务器和第二代理服务器;

所述第一代理服务器用于将所述封装标签发送至所述第二代理服务器;

所述第二代理服务器用于根据所述变量信息对所述待注册节点进行管理。

4. 根据权利要求3所述的系统,其特征在于,所述封装标签包括所述待注册节点的临时IP地址和临时端口信息,其中,所述临时IP地址和临时端口为所述待注册节点接入系统时,所述待注册节点上的第三代理服务器为所述待注册节点临时分配的。

基于Kubernetes系统的节点注册方法及系统

技术领域

[0001] 本发明实施例涉及大数据技术领域,尤其涉及一种基于Kubernetes系统的节点注册方法及系统。

背景技术

[0002] Kubernetes是Google开源的容器集群管理系统,其提供应用部署、维护、扩展机制等功能,利用Kubernetes能够方便地管理跨机器运行的容器化的应用,其主要功能包括如下几个方面:

[0003] (1) 使用容器引擎Docker对应用程序进行包装、实例化、运行等操作。

[0004] (2) 以集群的方式运行、管理跨机器的容器。

[0005] (3) 解决Docker跨机器容器之间的通讯问题。

[0006] 但是,在现有的Kubernetes系统中,节点的部署仍需要人工手动进行注册配置,效率较低。

发明内容

[0007] 本发明实施例提供一种基于Kubernetes系统的节点注册方法及系统,用以提高Kubernetes系统中节点的注册效率。

[0008] 本发明实施例第一方面提供一种基于Kubernetes系统的节点注册方法,该方法包括:

[0009] 接收待注册节点发送的封装标签,其中所述封装标签为所述待注册节点对所述待注册节点的节点信息进行Label标签封装获得的;

[0010] 根据所述封装标签,为所述待注册节点分配IP地址和端口,以使所述待注册节点根据所述IP地址和所述端口进行内部参数配置,完成所述待注册节点的注册。

[0011] 本发明实施例第二方面提供一种Kubernetes系统,该系统包括:

[0012] 接收单元,用于接收待注册节点发送的封装标签,其中所述封装标签为所述待注册节点对所述待注册节点的节点信息进行Label标签封装获得的;

[0013] 配置单元,用于根据所述封装标签,为所述待注册节点分配IP地址和端口,以使所述待注册节点根据所述IP地址和所述端口进行内部参数配置,完成所述待注册节点的注册。

[0014] 本发明实施例,通过接收待注册节点根据Label标签封装机制所生成的封装标签,根据该封装标签为待注册节点分配IP地址和端口,使得待注册节点能够根据该IP地址和端口完成内部参数的配置,实现了Kubernetes系统下节点的自动注册,而不需要像传统技术那样,还需要对节点的注册参数进行手动配置,提高了Kubernetes系统下节点的注册效率。

附图说明

[0015] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现

有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅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性的前提下,还可以根据这些附图获得其他的附图。

[0016] 图1为本发明实施例一提供的基于Kubernetes系统的节点注册方法的流程示意图;

[0017] 图2为本发明实施例二提供的基于Kubernetes系统的节点注册方法的流程示意图;

[0018] 图3为本发明实施例三提供的Kubernetes系统的结构示意图;

[0019] 图4为本发明实施例四提供的Kubernetes系统的结构示意图。

具体实施方式

[0020] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,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,都属于本发明保护的范围。

[0021] 本发明的说明书和权利要求书的术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤的过程或结构的装置不必限于清楚地列出的那些结构或步骤而是可包括没有清楚地列出的或对于这些过程或装置固有的其它步骤或结构。

[0022] 图1为本发明实施例一提供的基于Kubernetes系统的节点注册方法的流程示意图,该方法可以通过一Kubernetes系统来执行。如图1所示,本实施例提供的方法包括如下步骤:

[0023] 步骤S101、接收待注册节点发送的封装标签,其中所述封装标签为所述待注册节点对所述待注册节点的节点信息进行Label标签封装获得的。

[0024] 本实施例中所称的“待注册节点”为初次接入Kubernetes系统,且未在Kubernetes系统上注册的节点。

[0025] 可选的,本实施例中,Kubernetes系统接收到的待注册节点发送的封装标签,可以被具体为Label标签。

[0026] 实际应用中,待注册节点通过节点上的代理服务器接入Kubernetes系统,并通过该代理服务器分配临时的IP地址和端口。在待注册节点接入Kubernetes系统并获得代理服务器分配的临时IP地址和临时端口后,待注册节点根据Kubernetes系统的Label标签封装机制,对其自身的节点信息进行封装,形成封装标签。其中,待注册节点的节点信息包括待注册节点的节点标识、临时IP地址、临时端口信息、变量信息以及节点功能的说明信息。并且,这里需要说明的是:本实施例中节点信息的封装方法与现有的Label标签封装机制类似,在这里不再赘述。进一步的,待注册节点在封装形成封装标签后,根据该临时的IP地址和端口将封装标签发送到Kubernetes系统中以使Kubernetes系统根据该封装标签对其进行注册处理。

[0027] 步骤S102、根据所述封装标签,为所述待注册节点分配IP地址和端口,以使所述待注册节点根据所述IP地址和所述端口进行内部参数配置,完成所述待注册节点的注册。

[0028] 实际应用中,Kubernetes系统在接收到封装标签后对该封装标签进行解析处理获取待注册节点的节点信息,进一步的,在获得待注册节点的节点信息后,Kubernetes系统通过其内部的域名系统为该待注册节点分配该待注册节点在Kubernetes系统内的IP地址和通信端口,并将该IP地址和该通信端口的信息发送给该待注册节点,以使该待注册节点根据Kubernetes系统分配的IP地址和通信端口进行内部的参数配置。从而实现Kubernetes系统下的节点的自动化注册。

[0029] 本实施例,通过接收待注册节点根据Label标签封装机制所生成的封装标签,根据该封装标签为待注册节点分配IP地址和端口,使得待注册节点能够根据该IP地址和端口完成内部参数的配置,实现了Kubernetes系统下节点的自动注册,而不需要像传统技术那样,还需要对节点的注册参数进行手动配置,提高了Kubernetes系统下节点的注册效率。

[0030] 图2为本发明实施例二提供的基于Kubernetes系统的节点注册方法的流程示意图,如图2所示,本实施例在步骤S101之后还可以包括步骤S103:

[0031] 步骤S103、通过第一代理服务器将所述封装标签发送至第二代理服务器,以使所述第二代理服务器根据所述封装标签对所述待注册节点进行管理。

[0032] 实际应用中,第一代理服务器可以被具体为Kubernetes系统中的客户端代理服务器。当Kubernetes系统接收到待注册节点发送的封装标签后,通过第一代理服务器将该封装标签发送给第二代理服务器,使得第二代理服务器能够根据该封装标签对该待注册节点进行管理。

[0033] 具体的,在对该待注册节点进行管理时,可选的,第二代理服务器从封装标签中获取待注册节点的变量信息,其中,该变量信息包括变量的名称以及变量对应的数据参数。进一步的,第二代理服务器依据Kubernetes系统下的Label标签匹配算法来实现对待注册节点的管理。其中,Kubernetes系统下的Label标签匹配算法为现有技术在这里不再赘述。

[0034] 本实施例,通过接收待注册节点根据Label标签封装机制所生成的封装标签,根据该封装标签为待注册节点分配IP地址和端口,使得待注册节点能够根据该IP地址和端口完成内部参数的配置,实现了Kubernetes系统下节点的自动注册,而不需要像传统技术那样,还需要对节点的注册参数进行手动配置,提高了Kubernetes系统下节点的注册效率。

[0035] 图3为本发明实施例三提供的Kubernetes系统的结构示意图,如图3所示,本实施例提供的Kubernetes系统包括:

[0036] 接收单元11,用于接收待注册节点发送的封装标签,其中所述封装标签为所述待注册节点对所述待注册节点的节点信息进行Label标签封装获得的;

[0037] 配置单元12,用于根据所述封装标签,为所述待注册节点分配IP地址和端口,以使所述待注册节点根据所述IP地址和所述端口进行内部参数配置,完成所述待注册节点的注册。

[0038] 其中,所述封装标签包括所述待注册节点的临时IP地址和临时端口信息,其中,所述临时IP地址和临时端口为所述待注册节点接入系统时,所述待注册节点上的第三代理服务器为所述待注册节点临时分配的。

[0039] 本实施例能够用于执行如图1所示实施例的方法,其执行方式和有益效果类似,在这里不再赘述。

[0040] 图4为本发明实施例四提供的Kubernetes系统的结构示意图,如图4所示,本实施

例在图3所示结构的基础上,还包括:

[0041] 管理单元13,其中所述管理单元包括第一代理服务器131和第二代理服务器132;

[0042] 所述第一代理服务器131用于将所述封装标签发送至所述第二代理服务器132;

[0043] 所述第二代理服务器132用于根据所述封装标签对所述待注册节点进行管理。

[0044] 特别的,本实施例中,所述封装标签中包括所述待注册节点的变量信息,所述第二代理服务器132具体用于根据所述变量信息对所述待注册节点进行管理。

[0045] 本实施例能够用于执行如图2所示实施例的方法,其执行方式和有益效果类似,在这里不再赘述。

[0046] 最后应说明的是:以上各实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述各实施例对本发明进行了详细的说明,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,或者对其中部分或者全部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发明各实施例技术方案的范围。

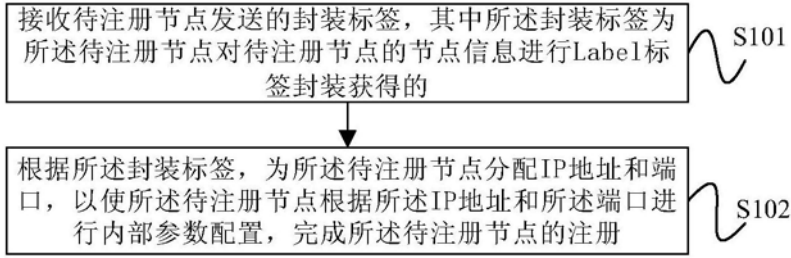


图1

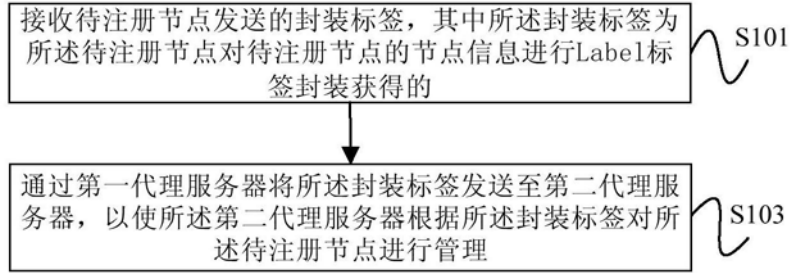


图2



图3

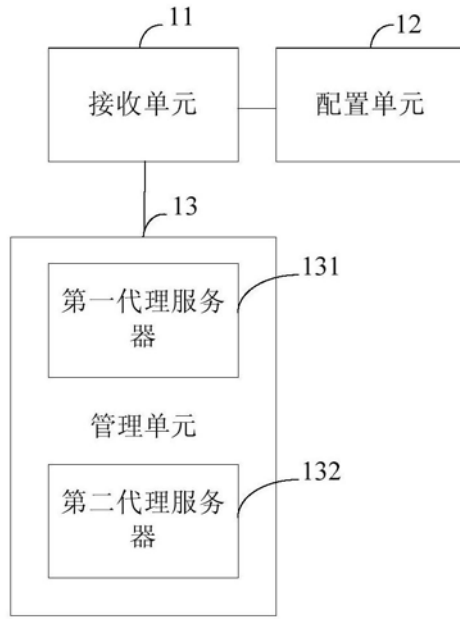


图4