



(12) 发明专利申请

(10) 申请公布号 CN 118051342 A

(43) 申请公布日 2024. 05. 17

(21) 申请号 202410333713.3

(22) 申请日 2024.03.22

(71) 申请人 中国人寿保险股份有限公司山东省分公司

地址 250032 山东省济南市历下区经十路11001号

(72) 发明人 单佳 姜润芝

(74) 专利代理机构 济南诚智商标专利事务有限公司 37105

专利代理师 李修杰

(51) Int. Cl.

G06F 9/50 (2006.01)

G06F 11/30 (2006.01)

H04L 12/28 (2006.01)

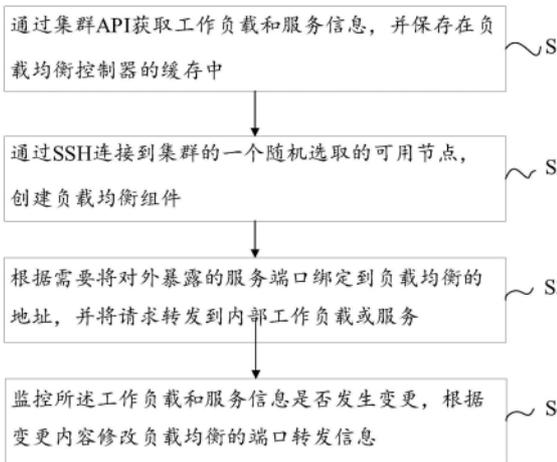
权利要求书2页 说明书6页 附图3页

(54) 发明名称

一种在局域网环境下创建k8s负载均衡的方法、系统、设备

(57) 摘要

本申请公开了一种在局域网环境下创建k8s负载均衡的方法、系统、设备,包括以下步骤:通过集群API获取工作负载和服务信息,并保存在负载均衡控制器的缓存中;通过SSH连接到集群的一个随机选取的可用节点,创建负载均衡组件;根据需要将对暴露的服务端口绑定到负载均衡的地址,并将请求转发到内部工作负载或服务;监听集群的事件,监控所述工作负载和服务信息是否发生变更,根据变更内容修改负载均衡的端口转发信息。该方案部署简单,成本低,无需对现有的集群做任何更改即可将集群内部的应用或者服务向集群外公开,同时该方案提供的负载均衡自带故障转移机制,可以提供高可用的服务。



1. 一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述方法包括以下步骤:

步骤S1:通过集群API获取工作负载和服务信息,并保存在负载均衡控制器的缓存中;

步骤S2:通过SSH连接到集群的一个随机选取的可用节点,创建负载均衡组件;

步骤S3:根据需要将对外暴露的服务端口绑定到负载均衡的地址,并将请求转发到内部工作负载或服务;

步骤S4:监控所述工作负载和服务信息是否发生变更,根据变更内容修改负载均衡的端口转发信息。

2. 根据权利要求1所述的一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述步骤S1之前,所述方法还包括:

通过ping命令间隔预设时间段检测各节点的可用性;

正常收到ping命令的响应时,标记该节点为可用;

当连续10次ping命令没有收到响应时,标记该节点为不可用,如果负载均衡部署在该节点,则执行故障转移。

3. 根据权利要求1所述的一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述工作负载和服务信息,具体为:命名空间、pod、service列表以及地址。

4. 根据权利要求1所述的一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述步骤S2,具体为:

通过随机算法从集群的节点中选取一个可用节点,在该节点的网卡上新绑定负载均衡的IP地址,并保存在数据库中。

5. 根据权利要求1所述的一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述步骤S3,具体为:

根据指定的负载均衡IP地址、对外暴露的端口和目标命名空间的Pod或服务信息,通过SSH登录到负载均衡所在的节点;

通过更新节点的iptables策略,实现负载均衡端口的流量到实际Pod或服务地址和端口的转发,同时将该策略缓存到本地。

6. 根据权利要求1所述的一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述步骤S4,具体为:

步骤S41:监控集群内工作负载和服务的变更信息,获取变更前和变更后的地址;

步骤S42:将变更后的地址添加到新的端口转发列表中,根据变更前的地址,删除对应的端口转发列表;

步骤S43:监听删除信息,获取删除的目的地址和端口,然后重复步骤S41-步骤S42,删除对应的端口转发列表。

7. 根据权利要求1所述的一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述方法还包括:

当监测到不可用节点重新可用时,负载均衡控制器通过SSH连接该重新可用的节点;

删除该节点上绑定的负载均衡IP地址和iptables中的转发策略,以避免冲突。

8. 根据权利要求7所述的一种在局域网环境下创建k8s负载均衡的方法,其特征在于,所述负载均衡控制器通过SSH连接该重新可用的节点的过程,具体为:

查找网络接口名称,修改网络接口的配置文件;
删除IPADDR1和PREFIX1所在的行,并重启网络服务。

9.一种在局域网环境下创建k8s负载均衡的系统,其特征在于,所述系统包括:
信息获取模块,用于通过集群API获取工作负载和服务信息,并保存在负载均衡控制器的缓存中;

监听模块,用于监听集群的事件,通过SSH连接到集群的节点;

创建模块,用于创建负载均衡,并绑定端口到服务;

执行模块,用于监控所述工作负载和服务信息是否发生变更,根据变更内容修改负载均衡的端口转发信息。

10.一种在局域网环境下创建k8s负载均衡的设备,其特征在于,所述设备包括:

至少一个处理器;以及,

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够完成如权利要求1-8任意一项所述的一种在局域网环境下创建k8s负载均衡的方法。

一种在局域网环境下创建k8s负载均衡的方法、系统、设备

技术领域

[0001] 本申请涉及Kubernetes集群部署技术领域,尤其涉及一种在局域网环境下创建k8s负载均衡的方法、系统、设备。

背景技术

[0002] Kubernetes是一个开源的容器编排平台,随着云原生技术体系的不断演进,kubernetes已经成为了容器编排上的事实上的标准。但是在非公有云环境下,kubernetes提供的将集群内服务暴露给外部的方案只有NodePort方案。

[0003] NodePort方案存在一些缺点,比如需要配置每个Pod所在的节点IP地址和端口号,配置过程相对复杂;此外NodePort依赖节点本身的网络地址,在节点不可用时,提供的外部访问地址也将变得不可用,基于NodePort方案需要额外的负载均衡配置方案。

发明内容

[0004] 本申请提供了一种在局域网环境下创建k8s负载均衡的方法、系统、设备一种在局域网环境下创建k8s负载均衡的方法,用以解决现有的NodePort方案需要额外的负载均衡配置方案且依赖节点本身的网络地址的技术问题。

[0005] 一方面,本申请提供了一种在局域网环境下创建k8s负载均衡的方法,所述方法包括以下步骤:步骤S1:通过集群API获取工作负载和服务信息,并保存在负载均衡控制器的缓存中;步骤S2:通过SSH连接到集群的一个随机选取的可用节点,创建负载均衡组件;步骤S3:根据需要将对外暴露的服务端口绑定到负载均衡的地址,并将请求转发到内部工作负载或服务;步骤S4:监控所述工作负载和服务信息是否发生变更,根据变更内容修改负载均衡的端口转发信息。

[0006] 在本申请的一种实现方式中,所述步骤S1之前,所述方法还包括:通过ping命令间隔预设时间段检测各节点的可用性;正常收到ping命令的响应时,标记该节点为可用;当连续10次ping命令没有收到响应时,标记该节点为不可用,如果负载均衡部署在该节点,则执行故障转移。

[0007] 在本申请的一种实现方式中,所述工作负载和服务信息,具体为:命名空间、pod、service列表以及地址。

[0008] 在本申请的一种实现方式中,所述步骤S2,具体为:通过随机算法从集群的节点中选取一个可用节点,在该节点的网卡上新绑定负载均衡的IP地址,并保存在数据库中;步骤S3,具体为根据指定的负载均衡IP地址、对外暴露的端口和目标命名空间的Pod或服务信息,通过SSH登录到负载均衡所在的节点;通过更新节点的iptables策略,实现负载均衡端口的流量到实际Pod或服务地址和端口的转发,同时将该策略缓存到本地。

[0009] 在本申请的一种实现方式中,所述步骤S4,具体为:步骤S41:监控集群内工作负载和服务的变更信息,获取变更前和变更后的地址;步骤S42:将变更后的地址添加到新的端口转发列表中,根据变更前的地址,删除对应的转发列表;步骤S43:监听删除信息,获

取删除的目的地址和端口,然后重复步骤S41-步骤S42,删除对应的端口转发列表。

[0010] 在本申请的一种实现方式中,所述方法还包括:当监测到不可用节点重新可用时,负载均衡控制器通过SSH连接该重新可用的节点;删除该节点上绑定的负载均衡IP地址和iptables中的转发策略,以避免冲突。

[0011] 在本申请的一种实现方式中,所述负载均衡控制器通过SSH连接该重新可用的节点的过程,具体为:查找网络接口名称,修改网络接口的配置文件;

[0012] 删除IPADDR1和PREFIX1所在的行,并重启网络服务。

[0013] 本申请实施例还提供了一种局域网环境下的k8s负载均衡系统,所述系统包括:信息获取模块,用于通过集群API获取工作负载和服务信息,并保存在负载均衡控制器的缓存中;监听模块,用于监听集群的事件,通过SSH连接到集群的节点;创建模块,用于创建负载均衡,并绑定端口到服务;执行模块,用于监控所述工作负载和服务信息是否发生变更,将变更后的地址添加到新的转发列表中。

[0014] 本申请实施例还提供了一种局域网环境下的k8s负载均衡设备,所述设备包括:至少一个处理器;以及与所述至少一个处理器通信连接的存储器;其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够完成前述的一种在局域网环境下创建k8s负载均衡的方法。

[0015] 本申请提供的一种在局域网环境下创建k8s负载均衡的方法、系统、设备。提供了一种局域网环境下的Kubernetes集群负载均衡的实现方案,该方案部署简单,成本低,无需对现有的集群做任何更改即可将集群内部的应用或者服务向集群外公开,同时该方案提供的负载均衡自带故障转移机制,可以提供高可用的服务。该控制器运行在单独的服务器中,可以根据用户的请求配置Kubernetes集群的节点的网络配置。

附图说明

[0016] 此处所说明的附图用来提供对本申请的进一步理解,构成本申请的一部分,本申请的示意性实施例及其说明用于解释本申请,并不构成对本申请的不当限定。在附图中:

[0017] 图1为本申请实施例提供的一种在局域网环境下创建k8s负载均衡的方法流程图;

[0018] 图2为本申请实施例提供的方案总体架构图;

[0019] 图3为本申请实施例提供的一种在局域网环境下创建k8s负载均衡的方法组成图;

[0020] 图4为本申请实施例提供的一种在局域网环境下创建k8s负载均衡的设备示意图。

具体实施方式

[0021] 为使本申请的目的、技术方案和优点更加清楚,下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然,所描述的实施例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。

[0022] 本申请实施例提供了一种在局域网环境下创建k8s负载均衡的方法、系统、设备,下面通过附图对本申请实施例提出的技术方案进行详细的说明。

[0023] 图1为本申请实施例提供的一种在局域网环境下创建k8s负载均衡的方法流程图。如图1所示,该方法主要包括以下步骤:

[0024] 步骤S1:通过集群API获取工作负载和服务信息,并保存在负载均衡控制器的缓存中;

[0025] 步骤S2:通过SSH连接到集群的一个随机选取的可用节点,创建负载均衡组件;

[0026] 步骤S3:根据需要将对外暴露的服务端口绑定到负载均衡的地址,并将请求转发到内部工作负载或服务;

[0027] 步骤S4:监控所述工作负载和服务信息是否发生变更,根据变更内容修改负载均衡的端口转发信息。

[0028] 在本申请实施例中,方案提供一种负载均衡控制器,该控制器运行在单独的服务器中,可以根据用户的请求配置Kubernetes集群的节点的网络配置,从而提供一种高可用的负载均衡方案实现集群内的服务向外部系统公开。方案总体架构如图2所示。

[0029] 负载均衡控制器主要包括以下功能:(1)、给指定的Kubernetes集群创建负载均衡。(2)、将指定的负载均衡的端口映射到集群内部的一个应用或服务的端口。(3)、定期检测负载均衡地址的可用性,在地址不可用时进行故障转移,实现高可用。

[0030] 系统的设计方案如下:(1)、负载均衡控制器保存Kubernetes集群各节点的IP地址、账号、口令、状态,期望设置的负载均衡的IP地址,和Kubernetes集群API的访问凭证。(2)、负载均衡控制器接受用户的请求,完成上述提供的功能。

[0031] 在本申请实施例中,负载均衡控制器通过ping命令定期检测各节点的可用性,当连续10次ping命令没有收到响应时标记该节点为不可用,如果负载均衡部署在该节点,需要执行故障转移。其中,可用性检测方案:假设目标节点的地址为10.50.2.3,直接执行命令ping 10.50.2.3并等待执行结果,如果1秒内未返回则重新执行ping命令,连续10次未收到结果则将目标节点标记为不可用。

[0032] 进一步地,负载均衡控制器初始会通过Kubernetes集群API获取命名空间、pod和service列表,并将名称、类型、地址等信息保存在负载均衡控制器的缓存中。负载均衡控制器与Kubernetes集群建立长连接,通过事件监听API获得集群内命名空间、Pod和Service的创建、变更、修改事件,同时更新本地缓存信息。

[0033] 获取集群命名空间列表的实现具体如下(采用python语言和kubernetes-client库):

```
[0034] import kubernetes
```

```
[0035] cluster=kubernetes.cluster.Cluster(cluster_name="cluster-name")
```

```
[0036] client=kubernetes.client.CoreV1Api(cluster)
```

```
[0037] namespaces=client.list_namespace()
```

[0038] 获取集群内Pod和Service列表的实现具体如下(采用python语言和kubernetes-client库):

```
[0039] client=kubernetes.client.CoreV1Api(cluster)
```

```
[0040] pods=client.list_pod_for_all_namespaces()
```

```
[0041] services=client.list_service_for_all_namespaces()
```

```
[0042] pods_dict=pods.to_dict()
```

```
[0043] services_dict=services.to_dict()。
```

[0044] 监听集群的事件,获取命名空间、Pod和Service的创建、变更、删除信息的实现具

体如下(采用python语言和kubernetes-client库):

```
from kubernetes import client, config, watch
# 配置 Kubernetes API 客户端
config.load_kube_config()
# 创建 Kubernetes API 客户端
api_instance = client.CoreV1Api()
[0045] def handle_event(event):
        print(event['type'], event['object'].metadata.name)
# 创建 Event Watcher, 注册事件类型和资源
w = watch.Watch()
for event in w.stream(api_instance.list_pod_for_all_namespaces):
        handle_event(event)。
```

[0046] 进一步地,负载均衡控制器通过SSH连接到集群的节点,以完成负载均衡的创建、配置等工作。

[0047] 通过SSH连接目标节点的方案:

[0048] ssh username:password@10.50.2.3。

[0049] 负载均衡的创建具体为:负载均衡控制器通过随机算法从集群的节点中选取一个可用节点,在该节点的网卡上新绑定负载均衡的IP地址,并保存在自己的数据库中。此时该节点拥有两个IP地址,一个用于Kubernetes集群管理,一个用于对外提供负载均衡的服务。

[0050] 本申请实施例中,随机算法具体为:random()*1000%节点个数。

[0051] 绑定负载均衡的IP地址的实现过程具体为(SSh连接后执行shell脚本):

[0052] #查找网络接口名称;

[0053] ip addr show match"inet 10.50.2.3";

[0054] #根据网络接口名称找到配置文件,并添加新的网络地址:

[0055] cat<<EOF>>/etc/sysconfig/network-scripts/ifcfg-〈网络接口名称〉;

[0056] IPADDR1=〈负载均衡IP〉;

[0057] PREFIX1=〈负载均衡IP的子网掩码〉;

[0058] EOF;

[0059] #重启网络服务;

[0060] systemctl restart network;

[0061] 重启后负载均衡的监听地址创建完成。

[0062] 本申请实施例中,绑定端口到服务:负载均衡控制器根据指定的负载均衡端口和目标命名空间的Pod或服务信息,通过SSH登录到负载均衡所在的节点,通过更新节点的iptables策略,实现负载均衡端口的流量到实际Pod或服务地址和端口的转发,同时将该策略缓存到本地。负载均衡控制器从本地缓存中找到目标命名空间的Pod或服务的IP

和端口信息,假设目标IP为172.17.2.2,端口80,为负载均衡指定的端口为8080。

[0063] 更新iptables配置,将负载均衡的80端口的流量重定向到172.17.2.2的80端口(SSH连接后执行shell脚本):

[0064] `iptables-t nat-A PREROUTING-p tcp-d<负载均衡IP>--dport 80-j DNAT--to-destination 172.17.2.2:80`

[0065] `iptables-t nat-A POSTROUTING-p tcp-d 172.17.2.2--dport 80-j SNAT--to-source<负载均衡IP>。`

[0066] 在本申请实施例中,在检测到负载均衡所在的节点不可用时,负载均衡控制器,重新选取一个节点,执行负载均衡创建操作,同时在该节点更新本地保存的iptables策略,同时更新本地缓存信息。此过程为故障转移。

[0067] 在本申请实施例中,监控变更信息,获取变更前和变更后的地址。将变更后的地址添加到新的转发列表中。根据变更前的地址,删除对应的转发列表(SSH连接后执行shell脚本):

[0068] `iptables-t nat-D PREROUTING-j DNAT--to-destination<变更前IP地址:端口号>`

[0069] `iptables-t nat-D POSTROUTING-j SNAT-d<变更前IP地址>--dport<变更前端口号>。`

[0070] 然后监听删除信息,获取删除的目的地址和端口,然后同步骤7.1,删除对应的转发列表。

[0071] 在本申请实施例中,故障恢复的过程具体为:当不可用节点重新可用时,负载均衡控制器通过SSH连接该节点,删除绑定的负载均衡IP地址和iptables中的转发策略,以避免冲突。找到网络接口名称,修改其配置文件/etc/sysconfig/network-scripts/ifcfg-<网络接口名称>,删除IPADDR1和PREFIX1所在的行,并重启网络服务。

[0072] 删除iptables中负载均衡相关的项目的过程为:根据前述变更过程以及本地缓存中的所有转发策略将所有策略删除。

[0073] 通过以上步骤,即可实现一个完整、高可用的Kubernetes负载均衡,用于将集群内应用和服务暴露给外部系统。

[0074] 以上是本申请实施例提供的一种在局域网环境下创建k8s负载均衡的方法,基于同样的发明构思,本申请实施例还提供了一种在局域网环境下创建k8s负载均衡的系统,图3为本申请实施例提供的一种局域网环境下的k8s负载均衡系统组成图,如图3所示,所述系统主要包括:信息获取模块301,用于通过集群API获取工作负载和服务信息,并保存在负载均衡控制器的缓存中;监听模块302,用于监听集群的事件,通过SSH连接到集群的节点;创建模块303,用于创建负载均衡,并绑定端口到服务;执行模块304,用于监控所述工作负载和服务信息是否发生变更,将变更后的地址添加到新的转发列表中。

[0075] 以上是本申请实施例提供的一种在局域网环境下创建k8s负载均衡的系统,基于同样的发明构思,本申请实施例还提供了一种在局域网环境下创建k8s负载均衡的设备,图4为本申请实施例提供的一种局域网环境下的k8s负载均衡设备示意图,如图4所示,该设备主要包括:至少一个处理器401;以及,与至少一个处理器通信连接的存储器402;其中,存储器402存储有可被至少一个处理器401执行的指令,指令被至少一个处理器401执行,以使至

少一个处理器301能够完成前述的一种在局域网环境下创建k8s负载均衡的方法。

[0076] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0077] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0078] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0079] 在一个典型的配置中,计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0080] 本申请中的各个实施例均采用递进的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于装置实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0081] 还需要说明的是,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

[0082] 以上所述仅为本申请的实施例而已,并不用于限制本申请。对于本领域技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本申请的权利要求范围之内。

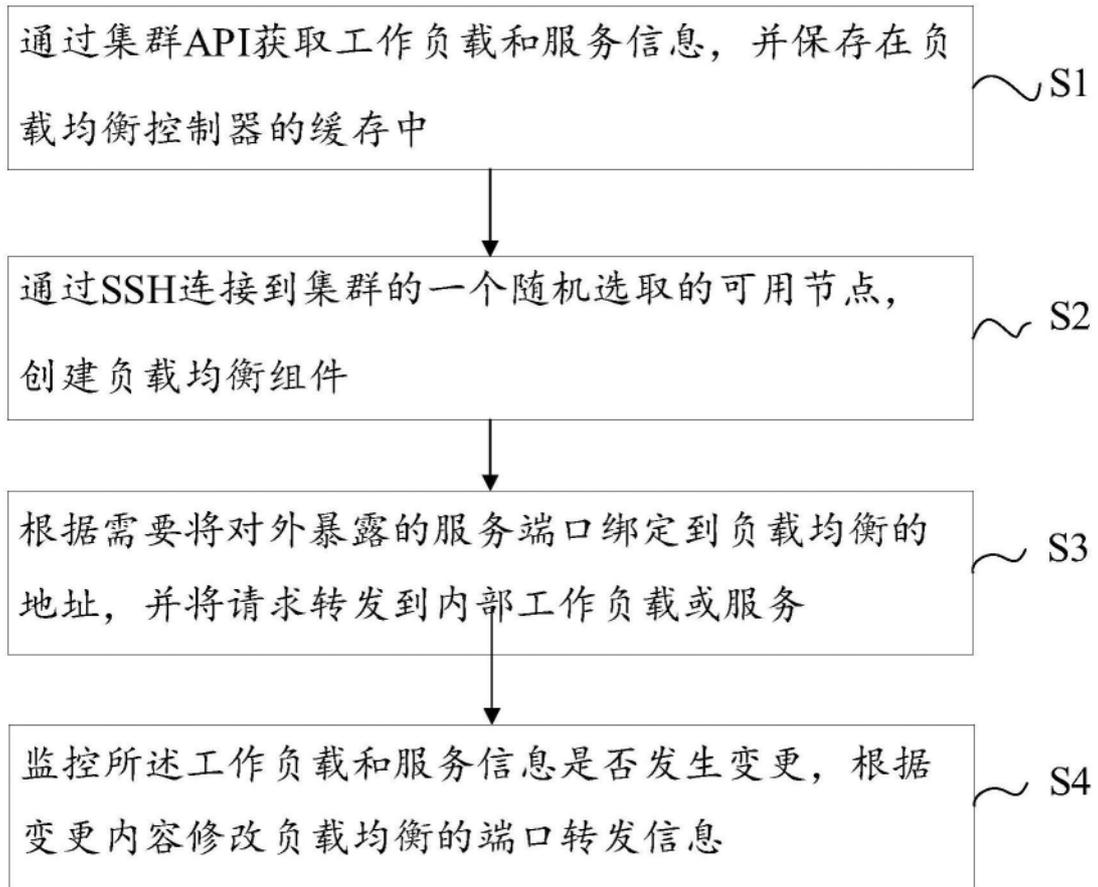


图1

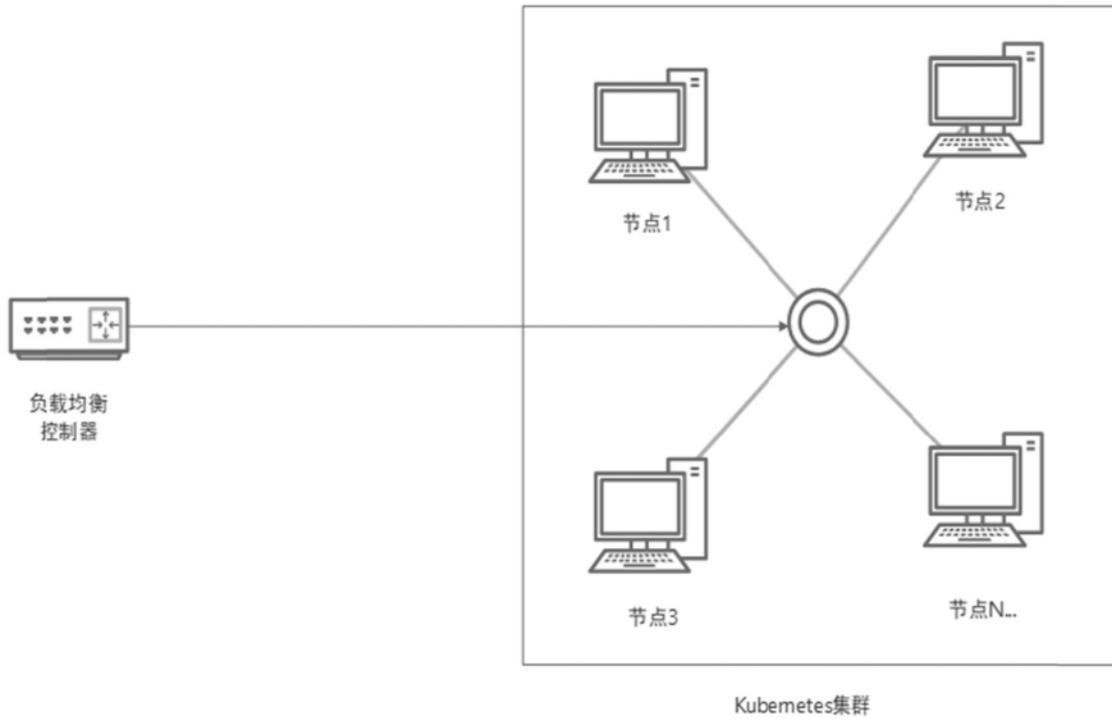


图2

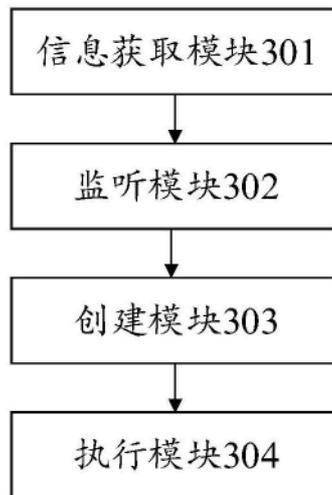


图3

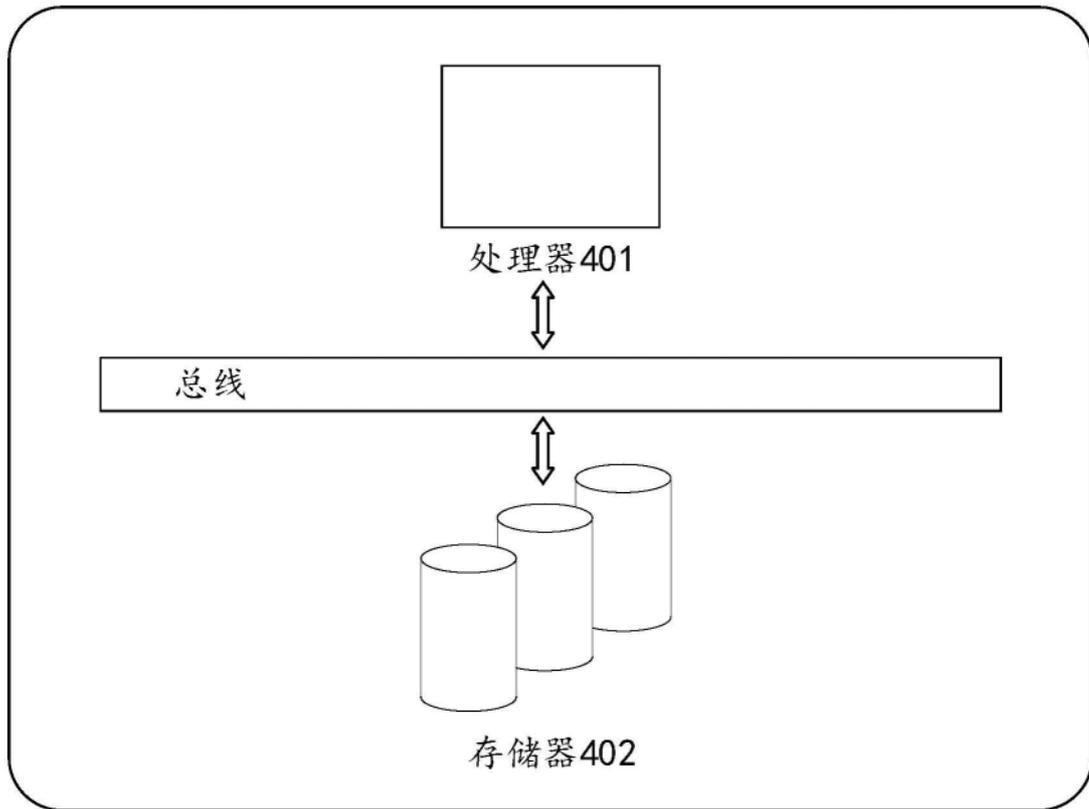


图4