



(21) 申请号 202110832741.6

(22) 申请日 2021.07.22

(65) 同一申请的已公布的文献号

申请公布号 CN 113687907 A

(43) 申请公布日 2021.11.23

(73) 专利权人 五八同城信息技术有限公司

地址 300450 天津市滨海新区经济技术开发区南港工业区综合服务区办公楼C座二层210-03室

(72) 发明人 王雷 张月 史忠伟

(74) 专利代理机构 北京润泽恒知识产权代理有限公司 11319

专利代理师 吕俊秀

(51) Int. Cl.

G06F 9/455 (2006.01)

(56) 对比文件

CN 105468362 A, 2016.04.06

CN 111506316 A, 2020.08.07

审查员 赵欣

权利要求书2页 说明书11页 附图2页

(54) 发明名称

一种应用处理方法、装置、电子设备及存储介质

(57) 摘要

本发明提供了一种应用处理方法、装置、电子设备及存储介质。该方法包括：响应于针对目标应用的镜像构建操作，向云平台发送镜像构建请求，所述镜像构建请求中携带所述目标应用对应的Dockerfile，所述Dockerfile中包含需要打包至镜像的多个目标文件的路径信息，所述多个目标文件中包含所述目标应用的应用程序包和用于启动所述应用程序包的启动脚本，以使所述云平台响应于所述镜像构建请求，通过运行所述Dockerfile，构建包含所述多个目标文件的镜像。如此，使得云平台对原本不兼容的软件框架的目标应用，也具有了镜像构建的逻辑和启动逻辑，提升了云平台的兼容性，提高了应用扩展能力。



1. 一种应用处理方法,其特征在于,包括:

响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,所述镜像构建请求中携带所述目标应用对应的Dockerfile,所述Dockerfile中包含需要打包镜像的多个目标文件的路径信息,所述多个目标文件中包含所述目标应用的应用程序包和用于启动所述应用程序包的启动脚本,以使所述云平台响应于所述镜像构建请求,通过运行所述Dockerfile,构建包含所述多个目标文件的镜像;

所述方法还包括:

响应于针对所述目标应用的镜像部署操作,向所述云平台发送镜像部署请求,以使所述云平台响应于所述镜像部署请求,通过所述启动脚本启动所述应用程序包,以进行镜像部署;所述目标应用的版本为新版本,所述镜像部署请求中携带第一节点分组的标识,所述第一节点分组中包括至少一个用于部署所述目标应用的镜像的第一节点;

在所述云平台已响应于所述镜像部署请求,基于所述第一节点分组的标识,对所述第一节点分组中所述第一节点完成所述目标应用的镜像部署之后,向所述云平台发送下线请求,所述下线请求中携带第二节点分组的标识,所述第二节点分组中包括至少一个已部署所述目标应用对应的旧版本的第二节点,以使所述云平台响应于所述下线请求,基于所述第二节点分组的标识,将所述第二节点分组中所述第二节点已部署的所述目标应用对应的旧版本下线。

2. 根据权利要求1所述的方法,其特征在于,所述应用程序包中包含来自第三方的代码包。

3. 根据权利要求2所述的方法,其特征在于,所述多个目标文件中还包含预处理文件;所述预处理文件用于对所述应用程序包中的代码包进行去重。

4. 根据权利要求1所述的方法,其特征在于,还包括:

响应于针对所述第一节点分组的扩容操作,确定所述云平台已经对所述第一节点分组中的第一节点完成所述目标应用的镜像部署且所述第一节点分组的资源满足第一预设条件,向所述云平台发送扩容请求,以使所述云平台响应于所述扩容请求,增加所述第一节点分组中的所述第一节点的数量以扩容。

5. 根据权利要求1所述的方法,其特征在于,还包括:

响应于针对所述目标应用的回滚操作,向所述云平台发送回滚请求,以使所述云平台响应于所述回滚请求,将所述第二节点分组中所述第二节点已部署的所述目标应用对应的旧版本重新上线,以及将所述第一节点分组中所述第一节点已部署的所述目标应用下线。

6. 一种应用处理装置,其特征在于,包括:

第一发送模块,用于响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,所述镜像构建请求中携带所述目标应用对应的Dockerfile,所述Dockerfile中包含需要打包镜像的多个目标文件的路径信息,所述多个目标文件中包含所述目标应用的应用程序包和用于启动所述应用程序包的启动脚本,以使所述云平台响应于所述镜像构建请求,通过运行所述Dockerfile,构建包含所述多个目标文件的镜像;

所述装置还包括:

第二发送模块,用于:响应于针对所述目标应用的镜像部署操作,向所述云平台发送镜像部署请求,以使所述云平台响应于所述镜像部署请求,通过所述启动脚本启动所述应用

程序包,以进行镜像部署;所述目标应用的版本为新版本,所述镜像部署请求中携带第一节点分组的标识,所述第一节点分组中包括至少一个用于部署所述目标应用的镜像的第一节点;

第三发送模块,用于在所述云平台已响应于所述镜像部署请求,基于所述第一节点分组的标识,对所述第一节点分组中所述第一节点完成所述目标应用的镜像部署之后,向所述云平台发送下线请求,所述下线请求中携带第二节点分组的标识,所述第二节点分组中包括至少一个已部署所述目标应用对应的旧版本的第二节点,以使所述云平台响应于所述下线请求,基于所述第二节点分组的标识,将所述第二节点分组中所述第二节点已部署的所述目标应用对应的旧版本下线。

7.一种应用处理系统,其特征在于,包括部署平台和云平台;

所述部署平台,用于响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,所述镜像构建请求中携带所述目标应用对应的Dockerfile,所述Dockerfile中包含需要打包镜像的多个目标文件的路径信息,所述多个目标文件中包含所述目标应用的应用程序包和用于启动所述应用程序包的启动脚本;

所述云平台,用于响应于所述镜像构建请求,通过运行所述Dockerfile,构建包含所述多个目标文件的镜像;

所述部署平台,还用于响应于针对所述目标应用的镜像部署操作,向所述云平台发送镜像部署请求;

所述云平台,还用于响应于所述镜像部署请求,通过所述启动脚本启动所述应用程序包,以进行镜像部署;所述目标应用的版本为新版本,所述镜像部署请求中携带第一节点分组的标识,所述第一节点分组中包括至少一个用于部署所述目标应用的镜像的第一节点;

所述部署平台,还用于在所述云平台已响应于所述镜像部署请求,基于所述第一节点分组的标识,对所述第一节点分组中所述第一节点完成所述目标应用的镜像部署之后,向所述云平台发送下线请求,所述下线请求中携带第二节点分组的标识,所述第二节点分组中包括至少一个已部署所述目标应用对应的旧版本的第二节点;

所述云平台,还用于响应于所述下线请求,基于所述第二节点分组的标识,将所述第二节点分组中所述第二节点已部署的所述目标应用对应的旧版本下线。

8.一种电子设备,其特征在于,包括:处理器、存储器及存储在所述存储器上并可在所述处理器上运行的计算机程序,所述计算机程序被所述处理器执行时实现如权利要求1至5中任一项所述的应用处理方法的步骤。

9.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如权利要求1至5中任一项所述的应用处理方法的步骤。

一种应用处理方法、装置、电子设备及存储介质

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种应用处理方法、装置、电子设备及存储介质。

背景技术

[0002] 云计算平台也称为云平台,是基于硬件资源和软件资源的服务,提供计算、网络和存储能力。相关技术中的一种云平台,是基于容器技术(例如Docker)打包,适合各种应用场景的一站式云化解决方案。Docker是一个开源的应用容器引擎,让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中,然后部署到机器上,也可以实现虚拟化容器。但是,该云平台可能并不兼容某些软件框架的应用,限制了应用扩展。

发明内容

[0003] 本发明实施例提供一种应用处理方法、装置、电子设备及存储介质,以解决相关技术中的问题。

[0004] 本发明的目的是通过以下技术方案实现的:

[0005] 第一方面,本发明实施例提供一种应用处理方法,包括:

[0006] 响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,所述镜像构建请求中携带所述目标应用对应的Dockerfile,所述Dockerfile中包含需要打包镜像的多个目标文件的路径信息,所述多个目标文件中包含所述目标应用的应用程序包和用于启动所述应用程序包的启动脚本,以使所述云平台响应于所述镜像构建请求,通过运行所述Dockerfile,构建包含所述多个目标文件的镜像。

[0007] 在一种可能的实施方式中,还包括:响应于针对所述目标应用的镜像部署操作,向所述云平台发送镜像部署请求,以使所述云平台响应于所述镜像部署请求,通过所述启动脚本启动所述应用程序包,以进行镜像部署。

[0008] 在一种可能的实施方式中,所述应用程序包中包含来自第三方的代码包。

[0009] 在一种可能的实施方式中,所述多个目标文件中还包含预处理文件;所述预处理文件用于对所述应用程序包中的代码包进行去重。

[0010] 在一种可能的实施方式中,所述目标应用的版本为新版本,所述镜像部署请求中携带第一节点分组的标识,所述第一节点分组中包括至少一个用于部署所述目标应用的镜像的第一节点;

[0011] 所述方法还包括:

[0012] 在所述云平台已响应于所述镜像部署请求,基于所述第一节点分组的标识,对所述第一节点分组中所述第一节点完成所述目标应用的镜像部署之后,向所述云平台发送下线请求,所述下线请求中携带第二节点分组的标识,所述第二节点分组中包括至少一个已部署所述目标应用对应的旧版本的第二节点,以使所述云平台响应于所述下线请求,基于所述第二节点分组的标识,将所述第二节点分组中所述第二节点已部署的所述目标应用对

应的旧版本下线。

[0013] 在一种可能的实施方式中,还包括:

[0014] 响应于针对所述第一节点分组的扩容操作,确定所述云平台已经对所述第一节点分组中的第一节点完成所述目标应用的镜像部署且所述第一节点分组的资源满足第一预设条件,向所述云平台发送扩容请求,以使所述云平台响应于所述扩容请求,增加所述第一节点分组中的所述第一节点的数量以扩容。

[0015] 在一种可能的实施方式中,还包括:

[0016] 响应于针对所述目标应用的回滚操作,向所述云平台发送回滚请求,以使所述云平台响应于所述回滚请求,将所述第二节点分组中所述第二节点已部署的所述目标应用对应的旧版本重新上线,以及将所述第一节点分组中所述第一节点已部署的所述目标应用下线。

[0017] 第二方面,本发明实施例提供一种应用处理装置,包括:

[0018] 第一发送模块,用于响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,所述镜像构建请求中携带所述目标应用对应的Dockerfile,所述Dockerfile中包含需要打包镜像的多个目标文件的路径信息,所述多个目标文件中包含所述目标应用的应用程序包和用于启动所述应用程序包的启动脚本,以使所述云平台响应于所述镜像构建请求,通过运行所述Dockerfile,构建包含所述多个目标文件的镜像。

[0019] 在一种可能的实施方式中,还包括:

[0020] 第二发送模块,用于:响应于针对所述目标应用的镜像部署操作,向所述云平台发送镜像部署请求,以使所述云平台响应于所述镜像部署请求,通过所述启动脚本启动所述应用程序包,以进行镜像部署。

[0021] 在一种可能的实施方式中,所述应用程序包中包含来自第三方的代码包。

[0022] 在一种可能的实施方式中,所述多个目标文件中还包含预处理文件;所述预处理文件用于对所述应用程序包中的代码包进行去重。

[0023] 在一种可能的实施方式中,所述目标应用的版本为新版本,所述镜像部署请求中携带第一节点分组的标识,所述第一节点分组中包括至少一个用于部署所述目标应用的镜像的第一节点;

[0024] 所述装置还包括:

[0025] 第三发送模块,用于在所述云平台已响应于所述镜像部署请求,基于所述第一节点分组的标识,对所述第一节点分组中所述第一节点完成所述目标应用的镜像部署之后,向所述云平台发送下线请求,所述下线请求中携带第二节点分组的标识,所述第二节点分组中包括至少一个已部署所述目标应用对应的旧版本的第二节点,以使所述云平台响应于所述下线请求,基于所述第二节点分组的标识,将所述第二节点分组中所述第二节点已部署的所述目标应用对应的旧版本下线。

[0026] 在一种可能的实施方式中,还包括:

[0027] 第四发送模块,用于响应于针对所述第一节点分组的扩容操作,确定所述云平台已经对所述第一节点分组中的第一节点完成所述目标应用的镜像部署且所述第一节点分组的资源满足第一预设条件,向所述云平台发送扩容请求,以使所述云平台响应于所述扩容请求,增加所述第一节点分组中的所述第一节点的数量以扩容。

[0028] 在一种可能的实施方式中,还包括:

[0029] 第五发送模块,用于响应于针对所述目标应用的回滚操作,向所述云平台发送回滚请求,以使所述云平台响应于所述回滚请求,将所述第二节点分组中所述第二节点已部署的所述目标应用对应的旧版本重新上线,以及将所述第一节点分组中所述第一节点已部署的所述目标应用下线。

[0030] 第三方面,本发明实施例提供一种应用处理系统,包括部署平台和云平台;

[0031] 所述部署平台,用于响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,所述镜像构建请求中携带所述目标应用对应的Dockerfile,所述Dockerfile中包含需要打包镜像的多个目标文件的路径信息,所述多个目标文件中包含所述目标应用的应用程序包和用于启动所述应用程序包的启动脚本;

[0032] 所述云平台,用于响应于所述镜像构建请求,通过运行所述Dockerfile,构建包含所述多个目标文件的镜像。

[0033] 第四方面,本发明实施例提供一种电子设备,包括:处理器、存储器及存储在存储器上并可在处理器上运行的计算机程序,该计算机程序被处理器执行时实现如第一方面中任一的应用处理方法的步骤。

[0034] 第五方面,本发明实施例提供一种计算机可读存储介质,该计算机可读存储介质上存储有计算机程序,该计算机程序被处理器执行时实现如第一方面中任一的应用处理方法的步骤。

[0035] 上述技术方案中的优点或有益效果至少包括:

[0036] 可以在检测到镜像构建操作后,向云平台发送镜像构建请求,该镜像构建请求中携带目标应用的Dockerfile,该Dockerfile中包含需要打包至镜像的多个目标文件的路径信息,多个目标文件中包含目标应用的应用程序包和用于启动应用程序包的启动脚本,如此,使得不兼容目标应用的软件架构的云平台基于Dockerfile具有了镜像构建逻辑以及基于启动脚本具有了应用程序包的启动逻辑,从而实现了对目标应用的兼容,提升了云平台的兼容性,提高了应用扩展能力。

[0037] 上述概述仅仅是为了说明书的目的,并不意图以任何方式进行限制。除上述描述的示意性的方面、实施方式和特征之外,通过参考附图和以下的详细描述,本发明进一步的方面、实施方式和特征将会是容易明白的。

附图说明

[0038] 图1是本发明实施例中的一种示例性的系统架构图;

[0039] 图2是本发明实施例提供的一种示例性的应用处理方法的流程图;

[0040] 图3是本发明实施例中的一种示例性的系统架构图;

[0041] 图4是本发明实施例提供的一种示例性的应用处理装置的结构图;

[0042] 图5是本发明实施例提供的一种示例性的电子设备的结构图。

具体实施方式

[0043] 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作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。

[0044] 云计算平台也称为云平台,是基于硬件资源和软件资源的服务,提供计算、网络和存储能力。相关技术中的一种云平台,是基于容器技术(例如Docker)打包,适合各种应用场景的一站式云化解决方案。Docker是一个开源的应用容器引擎,让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中,然后部署到机器上,也可以实现虚拟化容器。但是,该云平台可能并不兼容某些软件框架的应用,限制了应用扩展。

[0045] 例如,相关技术中的一种云平台可以支持java Web和远程过程调用(RPC,Remote Procedure Call)软件框架的应用部署,但是却不支持Spring Boot软件框架的应用部署,限制了更多应用的扩展。

[0046] 为解决上述技术问题,本发明实施例提供一种应用处理方法。

[0047] 图1是本发明实施例提供的一种示例性的系统架构图。本发明实施例提供的一种应用处理方法可以应用于图1所示的系统架构图中。如图1所示,该系统包括终端101和服务器102,该服务器102中设有云平台112和部署平台122。实际应用中,服务器可以通过部署平台122为终端101提供操作页面。当用户需要部署的应用的软件框架,云平台112不能兼容时,可以通过终端101的操作页面进行操作,相应的,服务器可以通过部署平台122执行本发明实施例提供的应用处理方法,该方法包括:响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,镜像构建请求中携带目标应用对应的Dockerfile,Dockerfile中包含需要打包镜像的多个目标文件的路径信息,多个目标文件中包含目标应用的应用程序包和用于启动应用程序包的启动脚本,以使云平台响应于镜像构建请求,通过运行Dockerfile,构建包含多个目标文件的镜像。下面对本发明实施例提供的应用处理方法进行详细地介绍。

[0048] 图2是本发明实施例提供的一种示例性的应用处理方法的流程图。如图2所示,本实施例提供的一种应用处理方法,至少包括如下步骤:

[0049] 步骤201、部署平台响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,镜像构建请求中携带目标应用对应的Dockerfile,Dockerfile中包含需要打包至镜像的多个目标文件的路径信息,多个目标文件中包含目标应用的应用程序包和用于启动应用程序包的启动脚本。

[0050] 步骤202、云平台响应于镜像构建请求,通过运行Dockerfile,构建包含多个目标文件的镜像。

[0051] 这里的目标应用是待处理的应用,该目标应用的软件架构是云平台本身不兼容的软件架构。

[0052] Dockerfile是一个用来构建镜像的文本文件,文本内容包含了一条条构建镜像所需的指令和说明。其中的指令引入基本镜像、指定工作目录、删除指定目录的所有内容文件、复制到ROOT下、解压文件、移除掉多余的压缩包、指定回工作目录。其中,复制到ROOT下,即COPY指令,命令格式可以是:COPY<src><dest>。COPY指令复制所指向的文件或目录,将它添加到新镜像中,复制的文件或目录在镜像中的路径是<dest>。<src>所指定的源可以有多个。这里的目标文件的路径信息即所指定的源。实际应用中,可以预先将目标文件存放到云平台的预设位置,并得到存放目标文件的路径信息。然后,构建好目标应用的Dockerfile并

保存。

[0053] 云平台响应于镜像构建请求,通过Docker引擎,运行Dockerfile,构建包含多个目标文件的镜像。构建过程中,可以基于目标文件的路径信息获取到目标应用的多个目标文件。

[0054] 本实施例中,可以在检测到镜像构建操作后,向云平台发送镜像构建请求,该镜像构建请求中携带目标应用的Dockerfile,该Dockerfile中包含需要打包至镜像的多个目标文件的路径信息,多个目标文件中包含目标应用的应用程序包和用于启动应用程序包的启动脚本,如此,使得不兼容目标应用的软件架构的云平台基于Dockerfile具有了镜像构建逻辑以及基于启动脚本具有了应用程序包的启动逻辑,从而实现了对目标应用的兼容,提升了云平台的兼容性,提高了应用扩展能力。

[0055] 在示例性实施例中,上述方法还可以包括:响应于针对目标应用的镜像部署操作,向云平台发送镜像部署请求,相应的,云平台响应于镜像部署请求,通过启动脚本启动应用程序包,以进行镜像部署。

[0056] 实际应用中,云平台响应于镜像部署请求,进行镜像部署,镜像部署过程中,由于构建的镜像中包含有启动脚本,因此,可以使得云平台能够通过启动脚本的启动逻辑启动应用程序包,从而完成应用镜像部署,提升了云平台的兼容性,提高了应用扩展能力。

[0057] 另外,通过简单的用户操作,就可以完成在云平台的应用部署,部署简单方便,降低了对部署人员的要求。

[0058] 在示例性实施例中,上述应用程序包中可以包含来自第三方的代码包。

[0059] 相关技术中,基于云平台提供了各种各样的应用,这些应用以云平台的内部规定格式开发的代码包实现,但是也有一些功能的应用,可能需要依赖第三方的代码包,这将会限制云平台应用的扩展。

[0060] 这里的代码包可以为jar包。本实施例中,由于额外向云平台提供了目标应用的Dockerfile、启动脚本,这些Dockerfile、启动脚本与目标应用的应用程序包是匹配的,因此即使应用程序包中包含了来自第三方的代码包,云平台也可以通过Dockerfile、启动脚本,正常运行,从而使得云平台具备了支持第三方的代码包的能力。

[0061] 实际应用中,可以通过上述操作页面,选择需要的第三方代码包。

[0062] 在示例性实施例中,上述多个目标文件中还可以包含预处理文件;该预处理文件用于对应用程序包中的代码包进行去重。

[0063] 由于应用程序包中包含了来自第三方的代码包,那么,有可能会与自己开发的代码包的部分有重复,如果有重复,会导致应用启动失败。本实施例中,通过在至少一个目标文件中预置了预处理文件,以对应用程序包中的代码包进行去重,降低了应用启动失败的风险。

[0064] 实际应用中,用于部署应用的对象可以称为节点。节点的数量可以是一个,也可以是多个。多个节点形成的集合可以称为集群。该集群中的多个节点可以进行分组,得到多个节点分组。

[0065] 实际应用中,可以进行节点操作,例如关于节点的重启、关闭、开启、通过(即节点的镜像测试通过)等。也可以进行分组操作,分组操作是对集群的节点分组进行的操作,例如关于节点分组的部署、扩容、回滚、上线完成等操作。

[0066] 对一个应用来说,可以在一个节点分组中部署应用,该一个节点分组中部署需要的版本的应用。也可以在两个节点分组中部署应用,其中一个节点分组用于部署新版本的应用,另一个节点分组用于部署旧版本的应用。节点分组中的每个节点的资源可以部署一个版本的应用,以节省资源。

[0067] 在示例性实施例中,上述目标应用的版本可以为新版本,镜像部署请求中携带第一节点分组的标识,第一节点分组中包括至少一个用于部署目标应用的第一节点。相应的,上述方法还可以包括:在云平台已响应于镜像部署请求,基于第一节点分组的标识,对第一节点分组中第一节点完成目标应用的镜像部署之后,向云平台发送下线请求,下线请求中携带第二节点分组的标识,第二节点分组中包括至少一个已部署目标应用对应的旧版本的第二节点,相应的,云平台响应于下线请求,基于第二节点分组的标识,将第二节点分组中第二节点已部署的目标应用对应的旧版本下线。

[0068] 其中,第一节点分组的标识是第一节点分组的身份标识。第二节点分组的标识是第二节点分组的身份标识。

[0069] 实际应用中,当第二节点分组的旧版本应用需要下线时,可以先对第一节点分组进行新版本的应用的部署,然后将第二节点分组的旧版本应用下线。本实施例中,用户只需要进行一次镜像部署操作,响应于该镜像部署操作,先向云平台发送携带第一节点分组的标识的镜像部署请求,在监测到云平台针对镜像部署请求完成响应之后,无需用户再次进行下线操作,可以自动向云平台发送携带第二节点分组的标识的下线请求,对第二节点分组的旧版本应用进行下线,从而加快了部署速度。

[0070] 可以理解的是,在确定对目标应用完成镜像部署之后,可以进行上线。具体的,可以将部署目标应用的节点分组的状态修改为上线 (ONLINE)。

[0071] 如果部署应用的节点分组有一个,那么,响应于上线完成操作,直接向云平台发送上线完成请求,相应的,云平台响应于上线完成请求,将上线完成操作所针对的节点分组已部署的目标应用上线。

[0072] 如果部署应用的节点分组有两个,响应于上线完成操作,确定上线完成操作所针对的节点分组满足第二预设条件,向云平台发送上线完成请求,相应的,云平台响应于上线完成请求,将上线完成操作所针对的节点分组已部署的目标应用上线。其中的第二预设条件可以是上线完成操作所针对的节点分组存在可用节点、上线完成操作所针对的节点分组的节点数量不少于另一节点分组的节点数量、以及另一节点分组已下线成功。或者,第二预设条件可以是上述另一节点分组没有可用节点且上线完成操作所针对的节点分组存在可用节点。

[0073] 扩容可以是指针对一个节点分组增加用于部署应用的节点的数量。示例性的,上述应用处理方法还可以包括:响应于针对第一节点分组的扩容操作,确定云平台已经对第一节点分组中的第一节点完成目标应用的镜像部署且第一节点分组的资源满足第一预设条件,向云平台发送扩容请求,相应的,云平台响应于扩容请求,增加第一节点分组中的第一节点的数量以扩容。

[0074] 扩容操作中可以增加节点的数量。其中的第一预设条件则是扩容操作所针对的节点分组剩余资源的CPU的核心数、内存数满足所增加节点的数量对CPU的核心数、内存数的需求。

[0075] 回滚是将应用恢复到上一版本,即旧版本。如果部署应用的节点分组只有一个,当上线的新版本的应用不符合要求时,可以按如下方式进行回滚:响应于针对目标应用的回滚操作,向云平台发送回滚请求,相应的,云平台响应于回滚请求,将回滚操作所针对的节点分组中的目标应用下线,然后将目标应用对应的旧版本重新上线。

[0076] 如果部署应用的节点分组有两个,如上述第一分组节点和第二分组节点,当上线的新版本的应用不符合要求时,可以按如下方式进行回滚:响应于针对目标应用的回滚操作,向云平台发送回滚请求,相应的,云平台响应于回滚请求,将第二节点分组中第二节点已部署的目标应用对应的旧版本重新上线,以及将第一节点分组中第一节点已部署的目标应用下线。这个过程中通过两个节点分组,可以维持一个版本的应用始终是上线状态,避免影响用户使用。

[0077] 由于应用是基于Docker部署,因此,上述可用节点是指节点对应的Docker实例启动,且在注册中心的状态是可用(UP)。

[0078] 注册中心是网络中的一个机构,可以核实用户数字认证请求然后告诉认证中心发放认证。在注册中心的状态为不可用,标记为DOWN,在注册中心的状态为可用,标记为UP。

[0079] 在实际应用中,用户可以根据自己的需求,进行上述各个操作,非常方便地完成应用部署。

[0080] 在示例性实施例中,向云平台提供的目标接口发送操作对应的请求。

[0081] 实际应用中,云平台提供的目标接口可以是镜像查询接口、镜像操作接口、集群查询接口、集群操作接口或者其它接口。

[0082] 其中,镜像查询接口为查询镜像构建的完成状态的接口。用户进行镜像构建操作之后,还可以进行镜像查询操作,当检测镜像查询操作时,可以向云平台提供的镜像查询接口发送镜像查询请求,使得云平台响应于镜像查询请求,查询镜像构建的完成状态。获取该镜像构建的完成状态并发送给终端以使得终端显示镜像构建的完成状态。如此,用户可以方便的了解到镜像构建的情况。

[0083] 镜像操作接口为进行镜像构建的接口。当检测镜像构建操作时,可以向云平台提供的镜像操作接口发送镜像构建请求。

[0084] 集群查询接口为查询集群的ID及对应的资源、环境变量等集群基础信息的接口。可以将集群的ID与用户操作所针对的集群业务类型建立映射关系。根据该映射关系可以确定用户操作对应的集群的ID。可以向云平台提供的集群查询接口发送集群查询请求,以使云平台响应于该集群查询请求返回集群基础信息。

[0085] 集群操作接口为进行镜像部署、扩容、回滚、上线完成等的接口。当检测到镜像部署操作、扩容操作、回滚操作或者上线完成操作时,可以向云平台提供的集群操作接口发送操作对应的请求。

[0086] 如此,直接基于云平台自身的接口对不兼容的软件框架的应用进行部署,实现更加简单,方便。

[0087] 下面以一个具体应用场景为例,对本发明实施例提供一种应用处理方法进行更加详细地说明。

[0088] 本实施例中,以服务器中一种本身可以支持java Web和RPC等软件框架但不支持Spring Boot软件框架的云平台为例,进行介绍。为实现基于该云平台的进行Spring Boot

软件框架的应用部署,服务器还提供了部署平台,该部署平台为进行用户操作的终端与云平台之间的中间件,衔接用户操作和云平台,基于云平台提供的接口(例如镜像查询接口、镜像操作接口、集群查询接口、集群操作接口以及其它接口),为用户提供方便的Spring Boot软件框架的应用部署,为用户提供构建镜像、镜像部署、扩容、回滚、上线完成等操作。所采用的系统架构整体包括部署平台、MySQL数据库、配置中心、云平台文件管理、远程字典服务(Redis,Remote Dictionary Server)集群。参见图2,示例出了具体的系统架构。

[0089] 其中,部署平台接入的容器类型也即所支持的集群业务类型,可以支持不同的集群业务类型,例如集群业务类型A、B、C、D、E。目标应用可以针对其中一个集群业务类型。

[0090] 部署平台可以设置实例应用部署管理模块、镜像管理模块、集群管理模块、日志模块。可以通过镜像管理模块,向云平台的镜像操作接口发送镜像构建请求,或者向云平台的镜像查询接口发送镜像查询请求。可以通过实例应用部署管理模块,向云平台的集群操作接口发送镜像部署请求、扩容请求、回滚请求和上线完成请求。可以通过集群管理模块,向云平台的集群查询接口发送集群查询请求。可以通过日志模块打印部署平台的运行日志。

[0091] 其中,MySQL数据库可以存储集群基础信息、镜像、应用程序包(例如jar包)等。

[0092] 配置中心,以Wconfig表示,可以存储Dockerfile等。

[0093] 云平台文件管理可以存储预处理脚本、启动脚本等。

[0094] Redis集群为缓存数据库,可以实现分布式锁和异步刷新任务。

[0095] 基于此,下面分别对应用部署的每个阶段进行说明。

[0096] 一、镜像构建:将镜像构建需要的Dockerfile文件存储在Wconfig,自定义的预处理脚本、启动脚本存储在云平台文件管理中,需要的jar包存储在MySQL数据库。用户可以通过操作页面选择合适的jar包版本后(第三方jar包可选),点击构建控件以进行镜像构建操作,部署平台检测到该镜像构建操作后,可以向云平台的镜像操作接口发送镜像构建请求,该镜像构建请求中携带Dockerfile。云平台接收到镜像构建请求后,可以从镜像构建请求中解析出Dockerfile,并运行Dockerfile,构建出包含jar包、自定义的预处理脚本、启动脚本的镜像。

[0097] 二、镜像部署:用户可以通过操作页面点击部署控件以进行镜像部署操作,部署平台检测到镜像部署操作后,向云平台的集群操作接口发送镜像部署请求。云平台接收到镜像部署请求后,通过启动脚本启动jar包,以进行镜像部署。

[0098] 用户操作可以为分组操作,也可以节点操作。分组操作主要关于节点分组的部署、扩容、回滚、上线完成等操作。节点操作主要关于节点的重启、关闭、开启、通过(即节点的镜像测试通过)。

[0099] 如果镜像部署操作为针对第一节点分组中的第一节点的操作,部署平台可以向云平台发送针对第一节点分组中的第一节点的镜像部署请求;部署平台在云平台已响应于镜像部署请求,对第一节点分组中的第一节点进行镜像部署之后,向云平台发送针对第二节点分组的第二节点的旧版本应用的下线请求。云平台响应于下线请求,将第二节点分组的第二节点的旧版本的应用下线。如此,无需用户再次进行下线操作,可以自动向云平台发送针对第二节点分组的旧版本的应用的下线请求,加快了部署速度。

[0100] 三、扩容:扩容的前提条件是,需要扩容的节点分组已部署完成,且分组剩余资源充足(例如CPU核心数、内存数满足要扩容的节点),即可进行扩容,扩容后会对需要扩容的

节点分组增加节点数量。

[0101] 另外,回滚和上线完成的过程可以参考以上相关实施例,此处不做赘述。

[0102] 通过本实施例提供的方案可以达到的效果如下:

[0103] 基于云平台提供的接口构建一套简单,方便的平台,为用户提供方便的Spring Boot软件框架的应用部署操作,并支持第三方人员部署各自的jar包。

[0104] 图4是本发明实施例提供的一种示例性的应用处理装置的结构图。如图4所示,本发明实施例提供一种应用处理装置400,包括:

[0105] 第一发送模块401,用于响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,镜像构建请求中携带目标应用对应的Dockerfile,Dockerfile中包含需要打包镜像的多个目标文件的路径信息,多个目标文件中包含目标应用的应用程序包和用于启动应用程序包的启动脚本,以使云平台响应于镜像构建请求,通过运行Dockerfile,构建包含多个目标文件的镜像。

[0106] 在一种可能的实施方式中,还包括:

[0107] 第二发送模块402,用于:响应于针对目标应用的镜像部署操作,向云平台发送镜像部署请求,以使云平台响应于镜像部署请求,通过启动脚本启动应用程序包,以进行镜像部署。

[0108] 在一种可能的实施方式中,应用程序包中包含来自第三方的代码包。

[0109] 在一种可能的实施方式中,多个目标文件中还包含预处理文件;预处理文件用于对应用程序包中的代码包进行去重。

[0110] 在一种可能的实施方式中,目标应用的版本为新版本,镜像部署请求中携带第一节点分组的标识,第一节点分组中包括至少一个用于部署目标应用的镜像的第一节点;

[0111] 上述装置还包括:

[0112] 第三发送模块403,用于在云平台已响应于镜像部署请求,基于第一节点分组的标识,对第一节点分组中第一节点完成目标应用的镜像部署之后,向云平台发送下线请求,下线请求中携带第二节点分组的标识,第二节点分组中包括至少一个已部署目标应用对应的旧版本的第二节点,以使云平台响应于下线请求,基于第二节点分组的标识,将第二节点分组中第二节点已部署的目标应用对应的旧版本下线。

[0113] 在一种可能的实施方式中,上述装置还包括:

[0114] 第四发送模块404,用于响应于针对第一节点分组的扩容操作,确定云平台已经对第一节点分组中的第一节点完成目标应用的镜像部署且第一节点分组的资源满足第一预设条件,向云平台发送扩容请求,以使云平台响应于扩容请求,增加第一节点分组中的第一节点的数量以扩容。

[0115] 在一种可能的实施方式中,上述装置还包括:

[0116] 第五发送模块405,用于响应于针对目标应用的回滚操作,向云平台发送回滚请求,以使云平台响应于回滚请求,将第二节点分组中第二节点已部署的目标应用对应的旧版本重新上线,以及将第一节点分组中第一节点已部署的目标应用下线。

[0117] 关于上述实施例中的装置,其中各个模块执行操作的具体方式已经在有关该方法的实施例中进行了详细描述,此处将不做详细阐述说明。

[0118] 本发明实施例还提供一种应用部署系统,包括部署平台和云平台;

[0119] 部署平台,用于响应于针对目标应用的镜像构建操作,向云平台发送镜像构建请求,镜像构建请求中携带目标应用对应的Dockerfile,Dockerfile中包含需要打包镜像的多个目标文件的路径信息,多个目标文件中包含目标应用的应用程序包和用于启动应用程序包的启动脚本;

[0120] 云平台,用于响应于镜像构建请求,通过运行Dockerfile,构建包含多个目标文件的镜像。

[0121] 关于上述实施例中的各平台,其中执行操作的具体方式已经在有关该方法的实施例中进行了详细描述,此处将不做详细阐述说明。

[0122] 本发明实施例提供的一种电子设备,包括:处理器、存储器及存储在存储器上并可在处理器上运行的计算机程序,该计算机程序被处理器执行时实现如以上任一实施例中的应用处理方法的步骤,且能达到相同的技术效果,为避免重复,这里不再赘述。

[0123] 图5是本发明实施例提供的一种示例性的电子设备的结构示意图。如图5所示,该电子设备可以包括:处理器501、通信接口502、存储器503和通信总线504,其中,处理器501,通信接口502,存储器503通过通信总线504完成相互间的通信。处理器501可以调用存储器503中的计算机程序,以执行以上任一实施例中的应用处理方法。

[0124] 本发明实施例还提供一种计算机可读存储介质,该计算机可读存储介质上存储有计算机程序,该计算机程序被处理器执行时实现如以上任一实施例中的应用处理方法的步骤,且能达到相同的技术效果,为避免重复,这里不再赘述。其中,的计算机可读存储介质,如只读存储器(Read-Only Memory,简称ROM)、随机存取存储器(Random Access Memory,简称RAM)、磁碟或者光盘等。

[0125] 需要说明的是,在本文中,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、物品或者装置不仅包括那些要素,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、物品或者装置所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0126] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的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(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端(可以是手机,计算机,服务器,空调器,或者网络设备等)执行本发明各个实施例的方法。

[0127] 上面结合附图对本发明的实施例进行了描述,但是本发明并不局限于上述的具体实施方式,上述的具体实施方式仅仅是示意性的,而不是限制性的,本领域的普通技术人员在本发明的启示下,在不脱离本发明宗旨和权利要求所保护的范围情况下,还可做出很多形式,均属于本发明的保护之内。

[0128] 本领域普通技术人员可以意识到,结合本发明实施例中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应

认为超出本发明的范围。

[0129] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0130] 在本申请所提供的实施例中,应该理解到,所揭露的装置和方法,可以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,所述单元的划分,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,机械或其它的形式。

[0131] 所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。

[0132] 另外,在本发明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。

[0133] 所述功能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,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,本发明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,或者网络设备等)执行本发明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、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0134] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

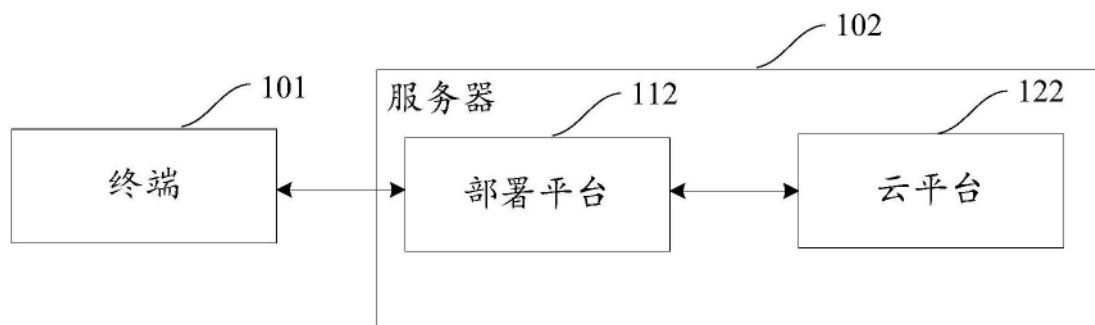


图1

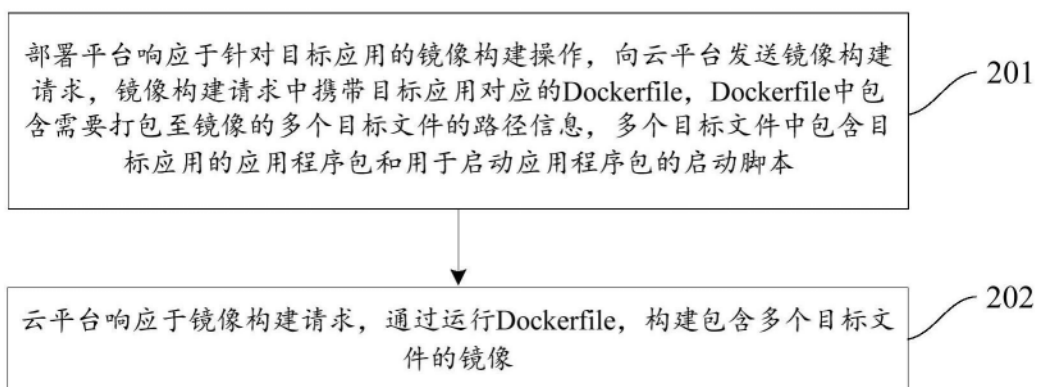


图2

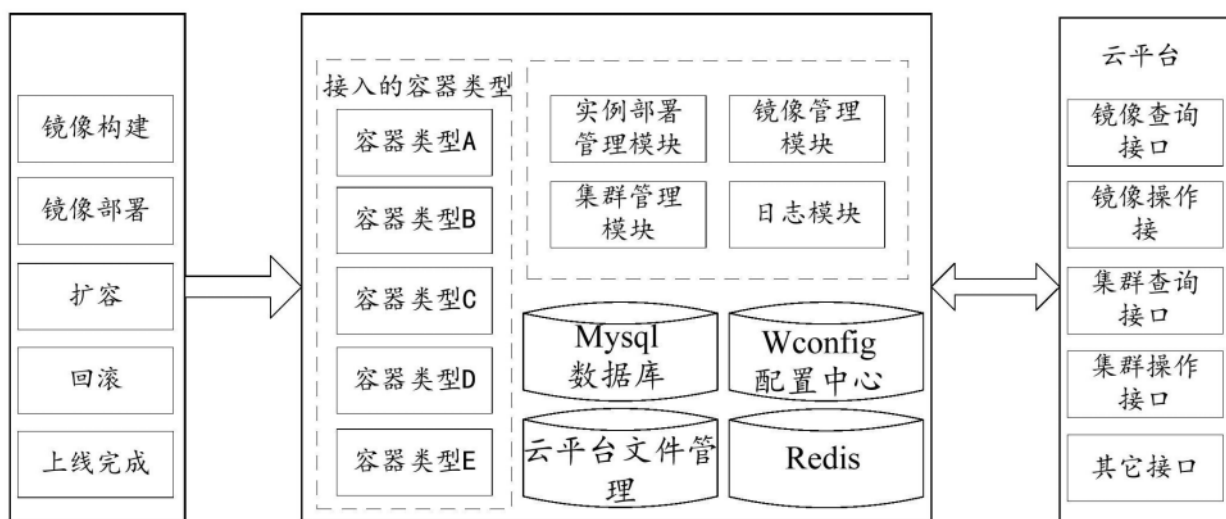


图3

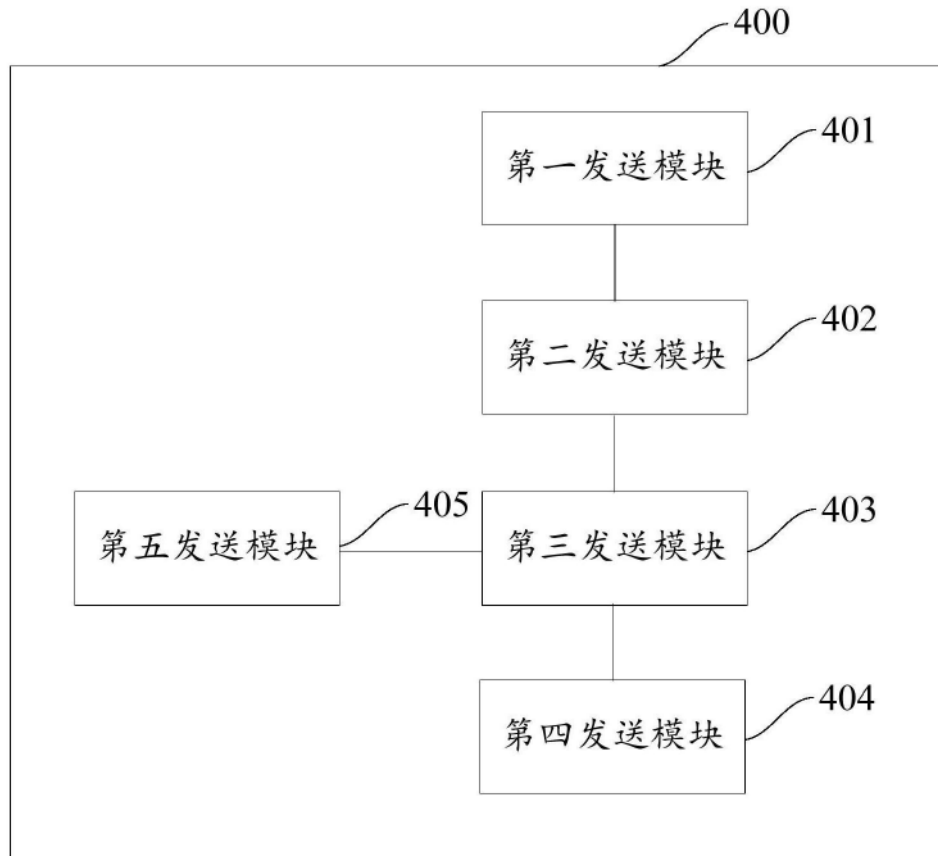


图4

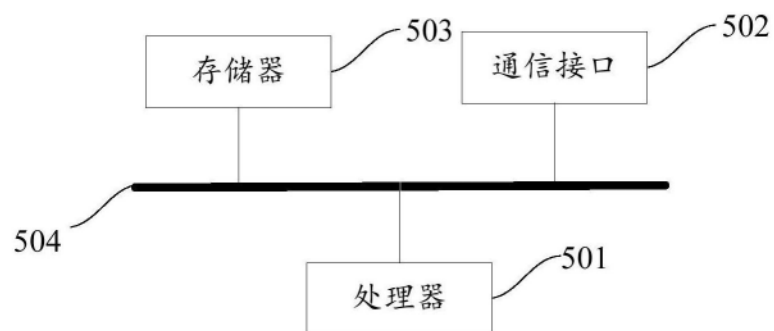


图5