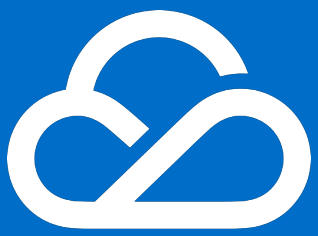
腾讯云容器服务

服务

产品文档



腾讯云

【版权声明】

【商标声明】

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

服务

服务概述

服务概述

服务是由多个相同配置的容器和访问这些容器的规则所组成。

服务类型

服务分为内网服务和外网服务。

集群内服务：访问方式不启用的，不绑定外网负载均衡，该服务只能在集群内部访问。

外网服务：访问方式选择了公网，自动绑定外网负载均衡，可通过外网负载均衡访问该服务。

内网服务：访问方式选择了内网，绑定内网负载均衡，可通过内网负载均衡访问该服务。

服务配置

用户在创建时可以自行配置服务，也可通过更新服务的方式来更新配置。

使用帮助

- 服务的基本操作
- 服务的生命周期
- 服务的访问方式设置
- 服务的资源限制设置
- 服务的运行命令和参数设置
- 服务的健康检查设置

服务的基本操作

创建服务

1. 登录 腾讯云容器服务控制台 。
2. 单击左侧导航栏中的【服务】，单击服务列表页的【新建】。

3. 设置服务的基本信息。
   - 服务名称：要创建的服务的名称，不超过 63 个字符。服务名称由小写字母、数字和 - 组成，且由小写字母开头，小写字母或数字结尾。
   - 所在地域：选择您部署服务的所在地域。
   - 运行集群：选择服务所要运行的集群。运行集群需要选择运行中和集群内有可用主机的集群。
   - 服务描述：创建服务的相关信息。该信息将显示在 服务信息 页面。

4. 设置数据卷。

   要指定容器挂载至指定路径时，单击【添加数据卷】。

   注意：

   源路径不指定时将默认分配临时路径。

   - 类型： 支持使用本地硬盘、云硬盘、NFS

     盘、配置文件四种类型的数据卷。相关详细介绍请参阅 容器服务数据卷使用说明。
   - 名称：数据卷的名称。
   - 路径：指定容器要挂载的路径。

5. 设置运行容器。

   - 名称 ：要创建容器的名称，不超过 63 个字符。
   - 镜像：单击【选择镜像】，可选择在我的镜像、我的收藏、TencentHub 镜像、DockerHub 镜像和其他镜像下创建服务。
   - 版本：容器服务默认选择版本。如果您需要使用镜像的其它版本，单击版本显示框选择。

6. 其他设置。
   - 实例数量：一个实例由相关的一个或多个容器构成。可单击 + 和 - 控制实例数量。
   - 服务访问方式
     ：服务的访问方式决定了这个服务的网络属性，不同访问方式的服务可以提供不同网络能力。提供的四种访问方式详细介绍请参阅 服务访问方式设置。

7. 单击【创建服务】完成服务创建。

更新实例数量

1. 单击容器服务控制台左侧导航栏中的【服务】，进入服务列表，单击【更新实例数量】。

2. 单击 + 和 - 控制新实例数量，设置完成后单击【确定】。

更新服务

1. 单击容器服务控制台左侧导航栏中的【服务】，进入服务列表，单击【更新服务】。

2. 单击 【开始更新】。

   提供两种更新方式。

   - 滚动更新：对实例进行逐个更新，这种方式可以让您不中断业务实现对服务的更新。
   - 快速更新：直接关闭所有实例，启动相同数量的新实例。

重新部署

重新部署是将服务下的容器重新部署一次，并重新拉取镜像。

1. 单击容器服务控制台左侧导航栏中的【服务】，进入服务列表，单击【更多】，单击【重新部署】。

2. 单击【确定】。

删除服务

1. 单击容器服务控制台左侧导航栏中的【服务】，进入服务列表，单击【更多】，单击【删除】。

2. 单击【确定】。

注意：

删除服务后该服务下所有实例和外网负载均衡将一并销毁，请提前备份好数据。

服务自动扩缩容

简介

服务自动扩缩容功能（又称 HPA）可以根据实例（pod）CPU
利用率等指标自动扩展，缩减服务的实例数量。
需要注意的是，自动扩缩容功能对应后台 HPA 组件的版本是 v2alpha1，并不支持 1.4.6 版本的 Kubernetes 集群。

使用方法

有下面三个入口可以设置服务的自动扩缩容：

- 创建/更新服务时设置：

- 更新服务实例数量时设置：

- 最大实例数，最小实例数：自动扩缩容功能能够调整的 Pod 数目区间。

- 触发策略指标：自动伸缩功能依赖的策略指标。
- CPU利用率：容器 CPU 使用量和 CPU 的 request 值比率。
- 内存利用率：容器内存使用量和内存的 request 值比率。
- 入带宽：POD 入带宽（Mb）。
- 出带宽：POD 出带宽（Mb）。

其中 CPU 利用率，内存利用率对应 Kubernetes 资源类指标；入带宽，出带宽对应 Kubernetes 自定义 Pod 类指标

如果需要使用资源类指标作为自动伸缩策略，需要在创建服务的时候设置容器对应的 request 值，否则不支持

伸缩算法

服务自动扩缩容后台组件会定期（30s）向腾讯云云监控拉取容器和 POD 的监控指标，然后根据该指标当前值

，当前副本数和该指标目标值计算出目标副本数，然后以该目标副本数作为服务的期望副本数，达到自动伸缩的目的。 比如当前有 2 个实例， 平均 CPU 利用率为 90%，服务自动伸缩设置目标 CPU 为 60%，
则自动调整实例数量为：90% * 2 / 60% = 3 个。
如果用户设置了多个弹性伸缩指标，HPA
会依据各个指标，分别计算出目标副本数，然后取最大的一个作为最终目标副本数

注意事项

- 为容器设置 CPU Request；
- 策略指标目标设置合理，如设置 70% 给容器和应用，预留 30% 的余量；
- 保持 Pod 和 Node 健康（避免 Pod 频繁重建）；
- 保证用户请求的负载均衡；
- HPA 在计算目标副本数时会有一个 10% 的波动因子，如果在波动范围内，HPA 并不会调整副本数目；
- 如果服务对应的deployment.spec.replicas值为0，弹性伸缩将不起作用；

Label 使用指引

Label 标签概述

Label 标签本质上是一对 key / value 被关联到对象上，对象可以是 Pod，Node 等 Kubernetes 资源。标签的使用我们倾向于能够标识对象的特殊属性，并且这些属性对用户而言是有意义的（例如：标志这个 Pod 是做什么的）。标签是可以用来划分特定组的对象（例如：所有 appA 的服务）。标签可以在创建一个对象的时候直接标记，也可以在后期随时修改，每一个对象可以拥有多个标签，但 key 值必须是唯一的。
腾讯云容器服务支持对服务实例进行 Label 标注。搜索服务时，也支持通过标签进行过滤。

服务中 Label 标签的使用

服务 Label 标签的展示

在服务列表页面和服务的详情页面可以查看服务的标签。由于腾讯云容器服务自身会对服务增加一定的标签，因此服务的标签分为系统标签（不可修改）和用户标签（可以修改）。
服务列表页展示服务的标签：

服务详情页展示的服务标签：

系统标签说明：

标签 Key 名称	标签 Value 意义
qcloud-app	服务名称

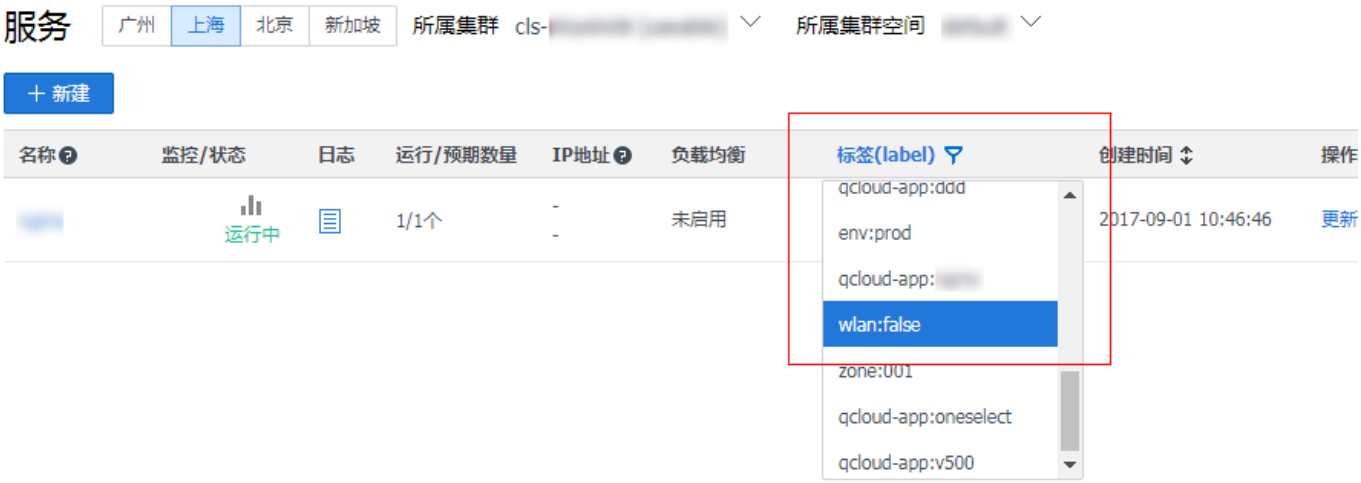
添加 / 删除 Label 标签

在服务列表页面或服务的详情列表，单击标签展示，即可以对标签进行编辑，编辑完成后保存新的标签。

> 注意：
> 系统标签不支持修改。

按 Label 标签过滤服务

在服务列表页面，支持按照标签搜索服务通过标签筛选，只展示标记了相应标签的服务。


服务
广州
上海
北京
新加坡
所属集群
所属集群空间
+ 新建
名称
监控/状态
日志
运行/预期数量
IP地址
负载均衡
标签(label)
创建时间
操作
运行中
1/1个
未启用
2017-09-01 10:46:46
更新
qcloud-app:ddd
env:prod
wlan:false
zone:001
qcloud-app:oneselect
qcloud-app:v500

服务的生命周期

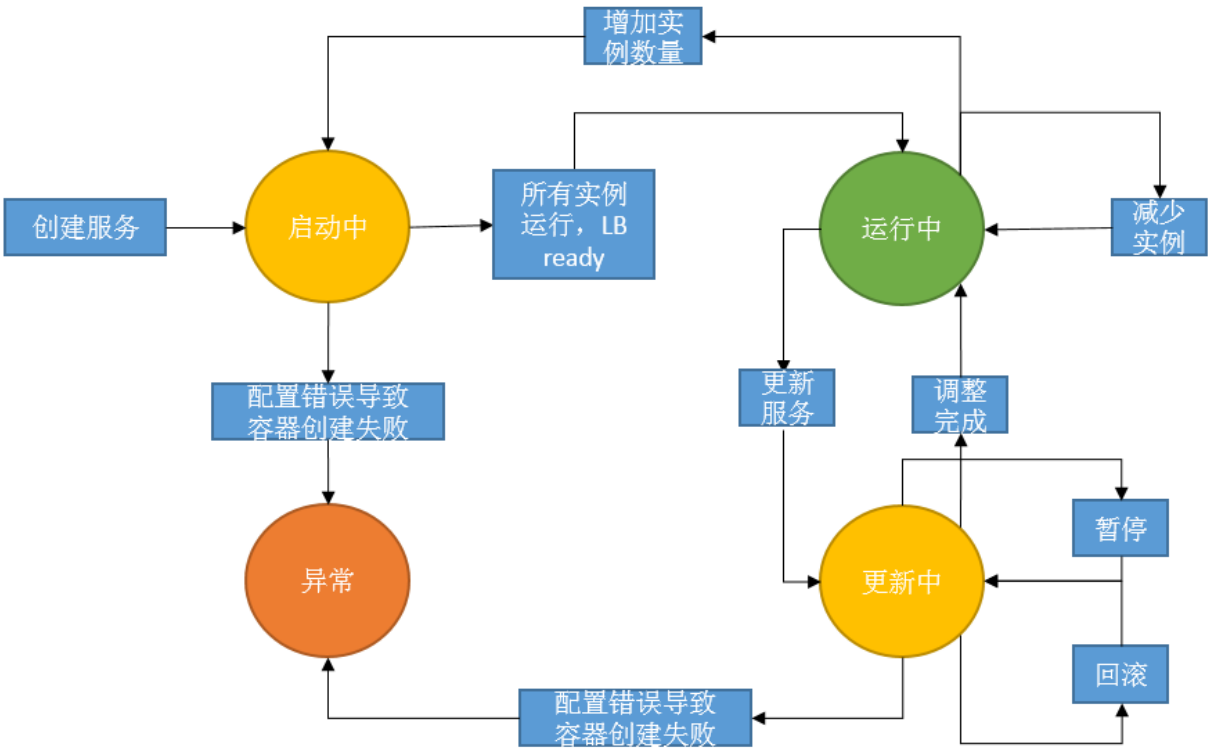
服务的生命周期

状态说明

状态	说明
启动中	服务正在创建，正在创建容器实例
运行中	服务正常运行，可对外提供服务
更新中	服务正在更新配置
异常	服务内存在异常的容器

状态流转图示

服务生命周期：服务状态之间转换如下图：



服务资源限制设置

请求（Request）与 限制（Limit）

Request：容器使用的最小资源需求，作为容器调度时资源分配的判断依赖。只有当节点上可分配资源量 >= 容器资源请求数时才允许将容器调度到该节点。但 Request 参数不限制容器的最大可使用资源值。
Limit： 容器能使用的资源最大值，设置为 0 表示使用资源无上限。

注意：
更多

Limit

和

Request

参数介绍，点击 查看详情。

CPU 限制说明

CPU 资源允许设置 CPU 请求和 CPU 限制的资源量，以核（U）为单位，允许为小数。

注意：

1. CPU Request 作为调度时的依据，在创建时为该容器在节点上分配 CPU 使用资源，称为“已分配 CPU” 资源。
2. CPU Limit 限制容器 CPU 资源的上限，设置为 0 表示不做限制（CPU Limit >= CPU Request）。

内存限制说明

内存资源只允许限制容器最大可使用内存量。以 MiB 为单位，允许为小数。

注意：

1. 由于内存资源为不可伸缩资源，在容器使用内存量超过内存 Request 时，容器就存在被 Kill 掉的风险。因此为了保证容器的正常运作限制 Request = Limit。
2. 内存 Request (= Limit)
   作为调度时的依据，在创建时为该容器在节点上分配内存使用资源，称为 “已分配内存” 资源。

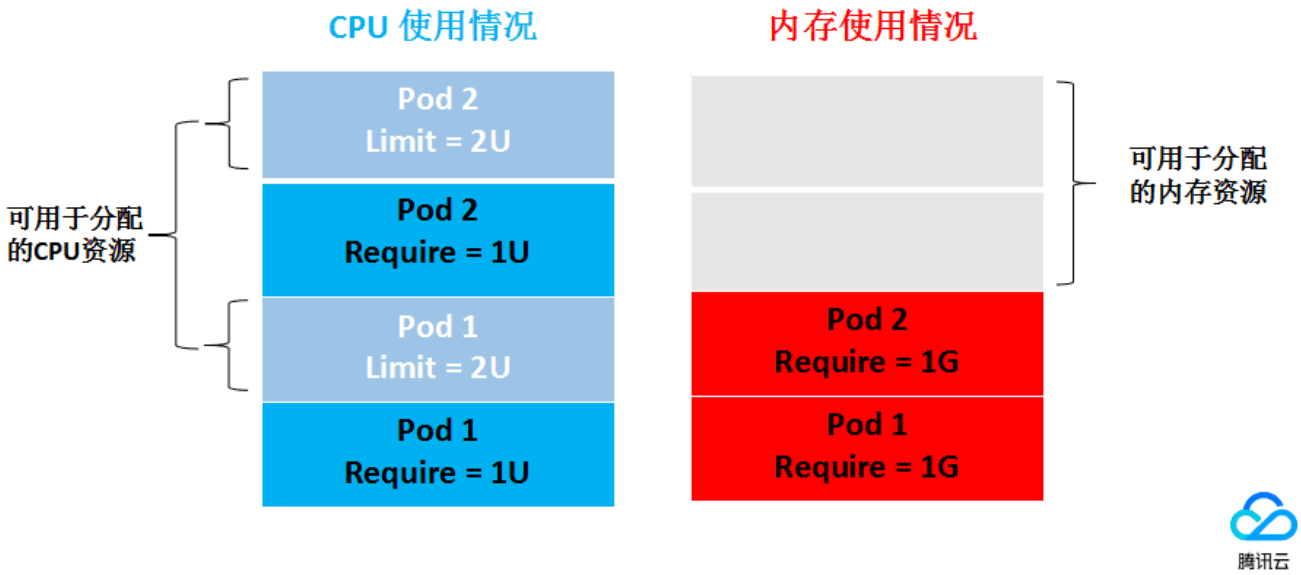
CPU 使用量 VS CPU 使用率

注意：

1. CPU 使用量为绝对值，表示实际使用的 CPU 的物理核数，CPU 资源请求和 CPU 资源限制的判断依据都是 CPU 使用量。
2. CPU 使用率为相对值，表示 CPU 的使用量与 CPU 单核的比值 (或者与节点上总 CPU 核数的比值)。

使用示例

一个简单的示例说明 Request 和 Limit 的作用，测试集群包括 1 个 4U4G 的节点、已经部署的两个 Pod (Pod1，Pod2)，每个 Pod 的资源设置为 (CPU Requst，CPU Limit，Memory Requst，Memory Limit) = (1U，2U，1G，1G)。(1.0 G = 1000 MiB)
节点上 CPU 和内存的资源使用情况如下图所示：



已经分配的 CPU 资源为：1U (分配 Pod1) + 1U (分配 Pod2) = 2U，剩余可以分配的 CPU 资源为 2U。
已经分配的内存资源为：1G (分配 Pod1) + 1G (分配 Pod2) = 2G，剩余可以分配的内存资源为 2G。
所以该节点可以再部署一个 (CPU Requst， Memory Requst) =(2U，2G)的 Pod 部署，或者部署 2 个 (CPU Requst， Memory Requst) = (1U，1G) 的 Pod 部署。

在资源限制方面，每个 Pod1 和 Pod2 使用资源的上限为 (2U，1G)，即在资源空闲的情况下，Pod 使用 CPU 的量最大能达到 2U。

服务资源限制推荐

CCS会根据你当前容器镜像的历史负载来推荐request与limit值，使用推荐值会保证的你容器更加平稳的运行，大大减小出现异常的概率。

推荐算法：
我们首先会取出过去7天当前容器镜像分钟级别负载，并辅以百分位统计第95%的值来最终确定推荐的Request,Limit为Request的2倍。

Request = Percentile(实际负载[7d],0.95)
Limit = Request * 2

如果当前的样本数量（实际负载）不满足推荐计算的数量要求，我们会相应的扩大样本取值范围，尝试重新计

算，例如：去掉镜像tag,namespace,serviceName等筛选条件。如经过多次计算后同样未能得到有效值，则推荐值为空。

推荐值为空：
在使用过程中，会发现有部分值暂无推荐的情况，可能是由于以下几点造成：

1. 当前数据并不满足计算的需求，我们需要待计算的样本数量（实际负载）大于1440个，即有一天的数据
2. 推荐值小于你当前容器已经配置的Request或者Limit

注意：

1. 由于推荐值是根据历史负载来计算的，原则上，容器镜像运行真实业务的时间越长，推荐的值越准确。
2. 使用推荐值创建服务，有可能会因为集群资源不足造成容器无法调度成功，在保存时一定要确认当前集群的剩余资源。
3. 推荐值是建议值，用户可以根据自己业务的实际情况做相应的调整。

设置容器运行命令

概述

创建服务时，我们通常都是通过镜像来指定实例中容器所运行的进程。在默认的情况下，镜像会运行默认的命令，如果我们想运行一个特定的命令或重写镜像的默认值，这里需要使用到以下三个设置：

- 工作目录（ workingDir ）：指定当前的工作目录。
- 运行命令（ command ）：控制镜像运行的实际命令。
- 命令参数（ args ）：传递给运行命令的参数。

工作目录说明

指定当前的工作目录，若不存在则自动创建，如果没有指定，则使用容器运行时的默认值。若镜像里面如果没指定 workdir，且在控制台未指定，workdir 默认为 " / "。

容器如何执行命令和参数

如何将 docker run 命令适配到腾讯云容器服务，点击查看 详情。

Docker 的镜像拥有存储镜像信息的相关元数据，如果不提供运行命令和参数，容器运行会运行镜像制作时提供的默认的命令和参数，Docker 原生定义这两个字段为 “ Entrypoint ” 和 " CMD "。详情可查看 docekr 的 Entrypoint 说明，CMD 说明。
如果在创建服务时填写了容器的运行命令和参数，将会覆盖镜像构建时的默认命令 " Entrypoint "、" CMD "，规则如下：

镜像 Entrypoint	镜像 CMD	容器的运行命令	容器的运行参数	最终执行
[ls]	[/home]	未设置	未设置	[ls / home]
[ls]	[/home]	[cd]	未设置	[cd]
[ls]	[/home]	未设置	[/data]	[ls / data]
[ls]	[/home]	[cd]	[/data]	[cd / data]

注意:
Docker entrypoint 对应容器服务控制台上的运行命令，Docker run 的 CMD 参数对应容器服务控制

台上的运行参数，当有多个运行参数时，在容器服务的运行参数中输入参数，每个参数单独一行。

服务健康检查设置

腾讯云容器集群内核基于 kubernetes。kubernetes
支持对容器进行周期性探测，根据探测结果来判断容器的健康状态，并执行额外的操作。

健康检查类别

健康检查分为两大类别：容器存活检查和容器就绪检查。

- 容器存活检查：该检查方式用于检测容器是否存活，类似于我们执行 ps 命令检查进程是否存在。如果容器的存活检查失败，集群会对该容器执行重启操作；若容器的存活检查成功则不执行任何操作。
- 容器就绪检查
  ：该检查方式用于检测容器是否准备好开始处理用户请求。一些程序的启动时间可能很长，比如要加载磁盘数据或者要依赖外部的某个模块启动完成才能提供服务。这时候程序进程在，但是并不能对外提供服务。这种场景下该检查方式就非常有用。如果容器的就绪检查失败，集群会屏蔽请求访问该容器；若检查成功，则会开放对该容器的访问。

健康检查方式

TCP 端口探测

TCP 端口探测的原理如下：
对于提供 TCP 通信服务的容器，集群周期性地对该容器建立 TCP
连接，如果连接成功，则证明探测成功，否则探测失败。选择 TCP
端口探测方式，必须指定容器监听的端口。比如我们有一个 redis 容器，它的服务端口是
6379，我们对该容器配置了 TCP 端口探测，指定探测端口为 6379，那么集群会周期性地对该容器的 6379
端口发起 TCP 连接，如果连接成功则证明检查成功，否则检查失败。

HTTP 请求探测

HTTP 请求探测针对的是提供 HTTP/HTTPS 服务的容器，集群周期性地对该容器发起 HTTP/HTTPS GET
请求，如果 HTTP/HTTPS response 返回码属于 200~399 范围，则证明探测成功，否则探测失败。使用
HTTP 请求探测必须指定容器监听的端口和 HTTP/HTTPS 的请求路径。
例如：提供 HTTP 服务的容器，服务端口为 80，HTTP 检查路径为

/health-check

，那么集群会周期性地对容器发起如下请求：

GET http://containerIP:80/health-check

。

执行命令检查

执行命令检查是一种强大的检查方式，该方式要求用户指定一个容器内的可执行命令，集群会周期性地在容器内执行该命令，如果命令的返回结果是 0 则检查成功，否则检查失败。
对于上面提到的 TCP 端口探测和 HTTP 请求探测，都可以通过执行命令检查的方式来替代：

- 对于 TCP 端口探测，我们可以写一个程序来对容器的端口进行 connect，如果 connect 成功，脚本返回 0，否则返回 -1。
- 对于 HTTP 请求探测，我们可以写一个脚本来对容器进行 wget

  wget http://127.0.0.1:80/health-check

  并检查 response 的返回码，如果返回码在 200~399 的范围，脚本返回 0，否则返回 -1。

  注意:

  - 必须把要执行的程序放在容器的镜像里面，否则会因找不到程序而执行失败。
  - 如果执行的命令是一个 shell 脚本，不能直接指定脚本作为执行命令，需要加上脚本的解释器。比如我们脚本是

    /data/scripts/health_check.sh

    ，那么我们使用执行命令检查时，指定的程序应该是

    sh /data/scripts/health_check.sh

。究其原因是集群在执行容器里的程序时，不在终端环境下。

其它公共参数

- 启动延时：单位秒。该参数指定了容器启动后，多久开始探测。例如启动延时设置为 5，那么健康检查将在容器启动 5 秒后开始。
- 间隔时间：单位秒。该参数指定了健康检查的频率。例如间隔时间设置成 10，那么集群会每隔 10s 检查一次。
- 响应超时：单位秒。该参数指定了健康探测的超时时间，对应到 TCP 端口探测、HTTP 请求探测、执行命令检查三种方式，分别表示 TCP 连接超时时间、HTTP 请求响应超时时间以及执行命令的超时时间。
- 健康阈值：单位次。该参数指定了健康检查连续成功多少次后，才判定容器是健康的。例如健康阈值设置成 3，则说明只有满足连续 3 次探测都成功才认为容器是健康的。

  注意:
  如果健康检查的类型为存活检查，那么健康阈值只能是 1，用户设置成其它值将被视为无效。因为只要探测成功一次，我们就能确定容器是存活的。

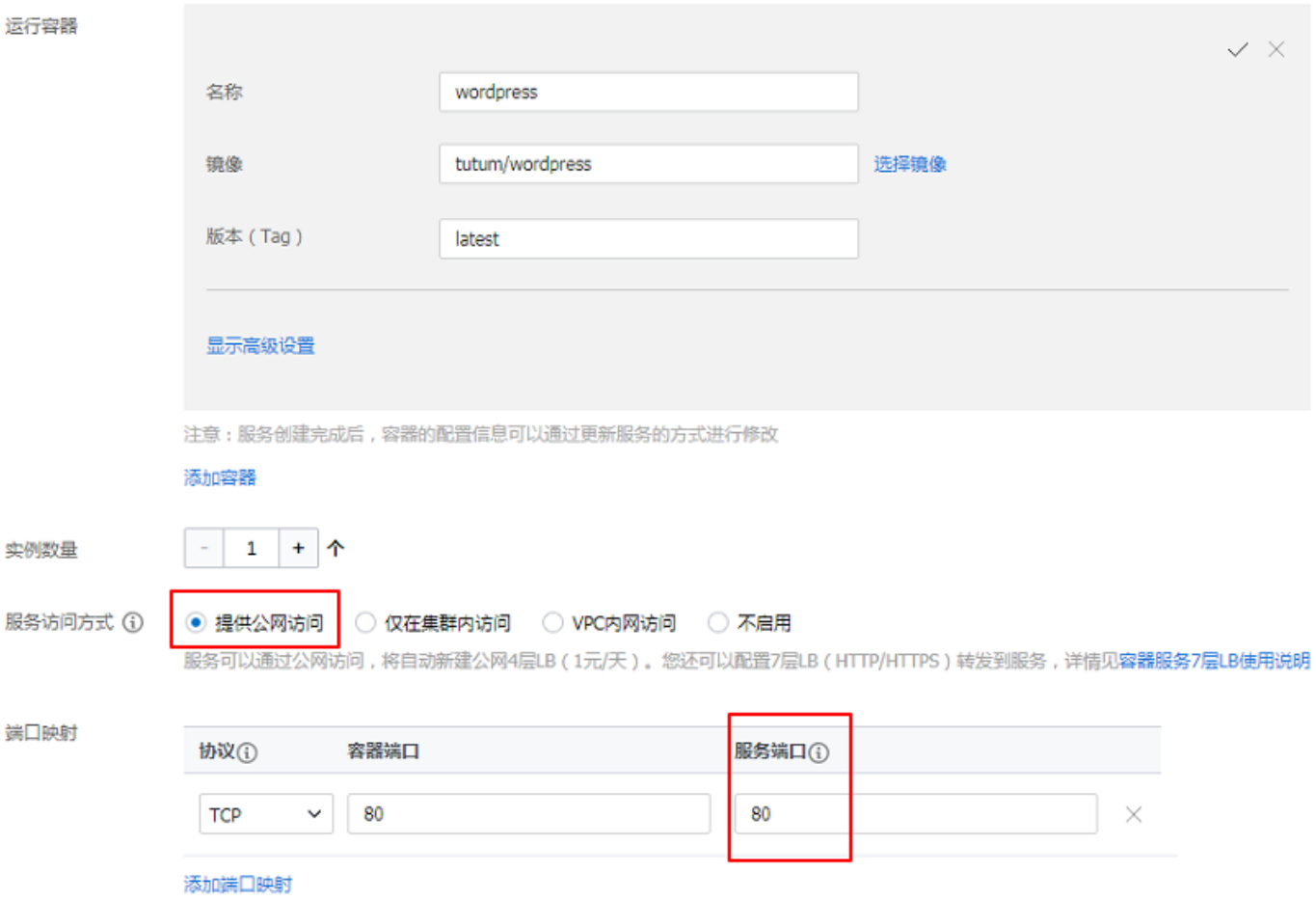
- 不健康阈值：单位次。该参数指定了健康检查连续失败多少次后，才判定容器是不健康的。例如不健康阈值设置成 3，只有满足连续 3 次都探测失败了，才认为容器是不健康的。

服务访问方式设置

服务的访问方式决定了这个服务的网络属性，不同访问方式的服务可以提供不同网络能力。腾讯云容器服务提供四种服务访问方式：提供公网访问、仅在集群内访问、VPC 内网访问和不启用。

提供公网访问

提供公网访问的服务将提供一个可以从 Internet
访问入口（公网负载均衡器）。选择公网访问的服务，可以直接被公网访问，web
前台类的服务可以选择公网访问，如 wordpress 前台服务。
例如创建一个可以提供公网访问的 wordpress 服务，在设置服务访问方式时选择 提供公网访问
。创建完成的服务可以通过 负载均衡 IP + 服务端口直接访问。



有关 wordpress 服务的具体创建操作，请参阅 单实例版 wordpress。

仅在集群内访问

提供集群内访问的服务将会提供一个可以被集群内其他服务或容器访问的入口（服务 IP），数据库类等服务如

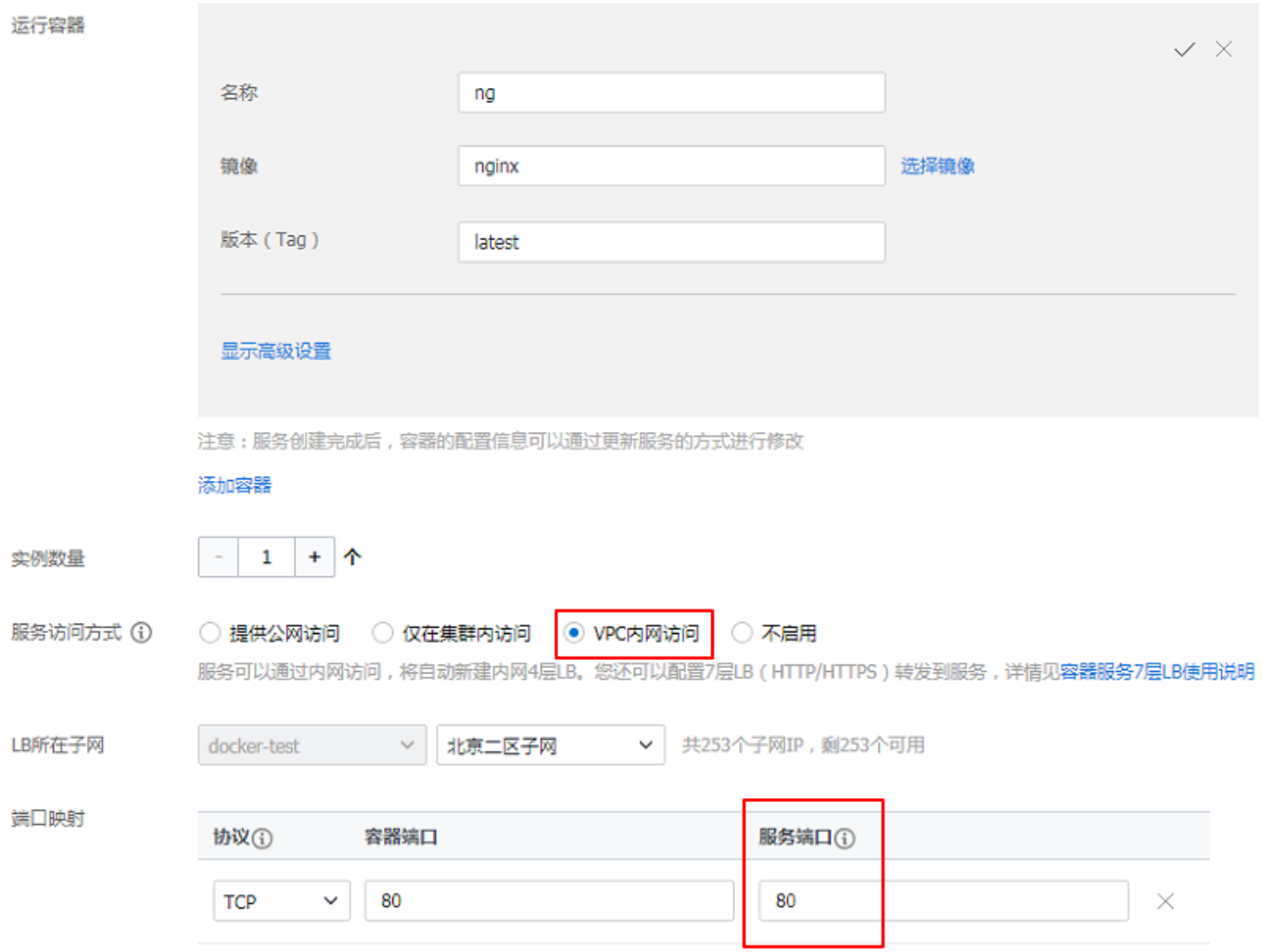
MySQL 可以选择集群内访问，以保证服务网络隔离。

例如创建一个仅在集群内访问的 MySQL 服务，在设置服务访问方式时选择 仅在集群内访问 。创建完成后的服务可以通过 服务 IP 或服务名称 + 服务端口 直接访问。

VPC 内网访问

VPC（私有网络）内网访问的服务将提供一个可以被集群所在 VPC 下的其他资源访问的入口（内网负载均衡器），需要被同一 VPC 下其他集群或 CVM（云服务器）访问的服务可以选择 VPC 内网访问的形式。

例如创建一个VPC内网访问的 MySQL 服务，在设置服务访问方式时选择 VPC 内网访问。创建完成后的服务可以通过 内网 LB 的 IP + 服务端口 直接访问。



不启用

选择不启用服务访问的服务，将不提供任何从前端服务访问到容器的入口，可用于使用自定义的服务发现或简单启用多个容器实例。

更多

除以上四种方式外还可以配置 7 层负载均衡（HTTP / HTTPS）转发到服务，详情请参阅 Ingress 转发设置。

更多关于集群访问方式原理介绍，请参阅 Kubenerters

中多种服务访问方式以及相应的安全组设置在腾讯云的落地实践。