

# 负载均衡

## 快速入门



腾讯云

## 【版权声明】

©2013-2026 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

## 【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100 或 95716。

# 文档目录

## 快速入门

[域名化负载均衡快速入门](#)

[负载均衡快速入门](#)

[IPv6 负载均衡快速入门](#)

[CentOS 下部署 Nginx](#)

[CentOS 下部署 Java Web](#)

# 快速入门

## 域名化负载均衡快速入门

最近更新时间：2025-05-19 18:04:32

腾讯云负载均衡支持 TCP、UDP、TCP SSL、QUIC、HTTP 和 HTTPS 协议，提供基于域名和 URL 路径的灵活转发能力。本文将引导您如何快速创建一个域名化负载均衡（CLB）实例，并将来自客户端的访问请求转发到后端服务器上（本文中的后端服务器以 CVM 为例）。

### 前提条件

1. 已创建两个云服务器实例（本文以 `rs-1` 和 `rs-2` 实例为例）。有关如何创建云服务器实例，请参考 [购买并启动云服务器实例](#)。
2. 已在云服务器上搭建后端服务。本文以 HTTP 转发为例，已在两台云服务器上部署 Nginx 服务器，并在 `rs-1` 和 `rs-2` 分别返回一个带有“Hello nginx! This is rs-1!” 和 “Hello nginx! This is rs-2!” 的 HTML，详情请参考 [CentOS 下部署 Nginx](#)。

#### 说明：

本文以标准账户类型（原“带宽上移账户”）为例。传统账户类型（原“带宽非上移账户”）的 CVM 必须购买公网带宽，因为当前的带宽属性在 CVM 上，而非 CLB 上。若您无法确定账户类型，请参见 [判断账户类型](#)。当前传统型负载均衡购买入口已关闭，如存量实例需要升级，请参考 [传统型负载均衡升级](#)。  
示例中后端服务器部署的服务返回值不同，实际场景中，为保持所有用户均有一致体验，后端服务器上一般是部署完全相同的服务。

### 使用步骤总览

1. [注册自有域名](#)
2. [购买负载均衡实例](#)
3. [配置负载均衡监听器](#)
4. [配置安全组](#)
5. [添加域名解析](#)
6. [验证负载均衡服务](#)

### 操作步骤

#### 步骤1：注册域名

域名注册是在互联网上建立服务的基础。

- 如果您已经在其他注册商拥有了自己的域名，您可以将域名转入腾讯云域名服务，详情请参见 [域名转入腾讯云](#)。
- 如果您还没有域名，您需要进行域名注册，详情请参见 [域名注册](#)。

## 步骤2：购买负载均衡实例

负载均衡实例成功购买后，系统将自动分配一个域名，您可以通过该域名使用负载均衡服务。

1. 登录腾讯云 [负载均衡服务购买页](#)。
2. 在负载均衡 CLB 购买页面，地域选择与云服务器相同的地域，实例类型选择负载均衡，网络类型选择公网，域名化负载均衡可根据使用场景选择一种内测方式进行申请：
  - 通过申请 [过渡模式内测](#)，在创建时选择 IP 模式。其中，动态 IP 模式为升级后的负载均衡实例。
  - 通过申请 [一键更新模式内测](#)，默认创建升级后的负载均衡实例。更多参数详情请参考 [产品属性选择](#)。
3. 单击立即购买，完成付款。
4. 返回实例管理页面，选择对应的地域即可看到新建的实例。

## 步骤3：配置负载均衡监听器

负载均衡监听器通过指定协议及端口来负责实际转发。本文以负载均衡转发客户端的 HTTP 请求配置为例。更多有关负载均衡监听器的更多内容，请参见 [负载均衡监听器概述](#)。

### 配置 HTTP 监听协议和端口

当客户端发起请求时，负载均衡会根据监听的前端协议与端口接收请求并向后端服务器转发请求。

1. 登录 [负载均衡控制台](#)。
2. 在实例管理页面中，找到目标负载均衡实例，单击配置监听器。
3. 在监听器管理页签中的“HTTP/HTTPS 监听器”区域下，单击新建。

安全防护：一键免费开通Web应用防火墙，为您的网站和APP提供安全防护。

温馨提示：当您配置了自定义重定向策略，原转发规则将进行修改。

## HTTP/HTTPS监听器 (已配置0个)

**新建**

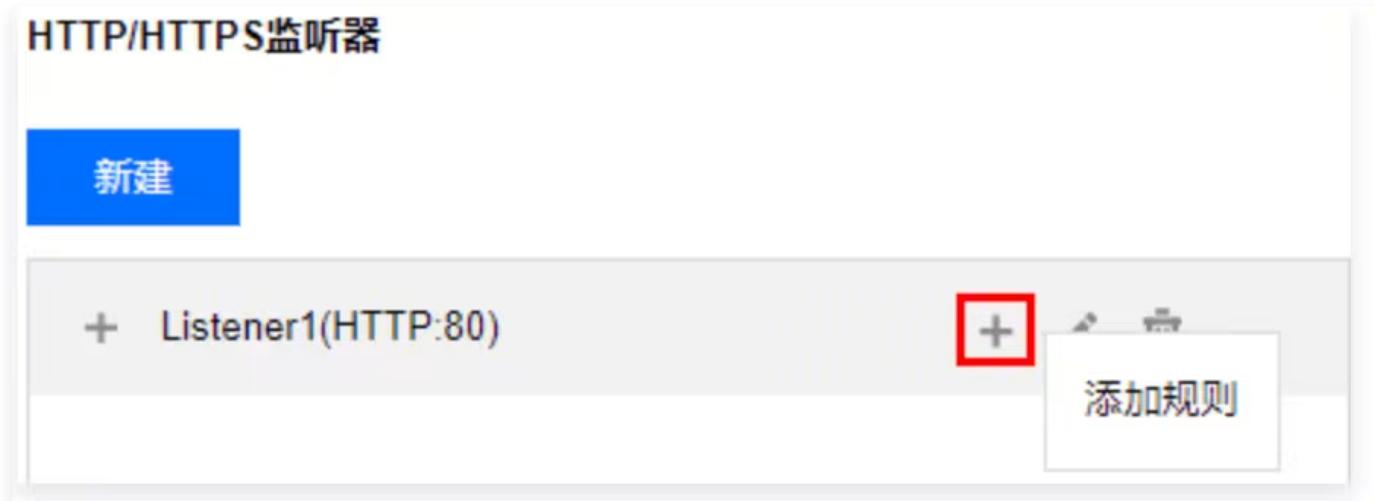
4. 在创建监听器对话框中，配置以下内容，配置完成后，单击提交。

- 监听器名称，名称可输入60个字符，允许英文字母、数字、中文字符、“-”、“\_”、“.”。
- 监听协议端口。例如 `HTTP: 80`。

### 配置监听器的转发规则

当客户端请求时，负载均衡会根据配置的监听器转发规则进行请求转发。

1. 在监听器管理页签中，选中刚才新建的监听器，单击 **+ 添加规则**。



2. 在创建转发规则对话框的“基本配置”页签中，配置域名、URL 路径和均衡方式。配置完成后，单击**下一步**。

- **域名**：您的后端服务所使用的域名，本例使用 `www.example.com`。详情请参见 [配置负载均衡的转发域名](#)。
- **默认域名**：当客户端请求没有匹配本监听器的任何域名时，CLB 会将请求转发给默认域名（`default server`），每个监听器只能配置一个默认域名。若该监听器没有配置默认域名时，CLB 会将请求转发给第一个域名。本例不配置。
- **URL 路径**：您的后端服务的访问路径，本例使用 `/image/`。
- **均衡方式**：选择“加权轮询”。详情请参见 [均衡方式](#)。

## 创建转发规则

X

1 基本配置 &gt; 2 健康检查 &gt; 3 会话保持

域名① [新增域名](#)默认域名  启用

当客户端请求没有匹配本监听器的任何域名时, CLB会将请求转发给默认域名 (Default Server), 每个监听器只能配置且必须配置一个默认域名, [详情](#)

URL路径① 均衡方式① 

WRR 根据新建连接数来调度, 权重越高的后端服务器被轮询到的概率越高

获取客户端IP  已启用Gzip压缩  已启用①后端目标组① [关闭](#)[下一步](#)

3. 在健康检查页签, 开启健康检查, 检查域名和检查路径使用默认的转发域名和转发路径, 完成后单击下一步。

## 创建转发规则



基本配置



健康检查



会话保持

健康检查



帮助您自动检查并移除异常的服务器端口。请注意检查在后端RS上配置的安全策略，需放通健康探测源IP。若未放通，将导致健康探测失败。

健康探测源IP



100.64网段



负载均衡VIP

检查协议



TCP



HTTP

检查域名



不填则默认为转发域名

检查路径



后端服务器的根目录



/

显示高级选项

上一步

下一步

## 4. 在会话保持页签，关闭会话保持，单击提交。

**说明：**

- 转发规则：一个监听器可配置多个域名，一个域名可配置多条URL路径，选中监听器或域名，单击+，即可创建新的规则。
- 会话保持：若关闭会话保持功能，选择轮询的方式进行调度，则请求会被依次分配到不同后端服务器上；若开启会话保持功能，或关闭会话保持功能但选择ip\_hash的调度方式，则请求会被持续分配到同一台后端服务器上。

## 为监听器绑定后端云服务器

当客户端请求时，负载均衡将请求转发到监听器绑定的云服务器上进行处理。

### 1. 在监听器管理页面，单击+展开刚才创建的监听器，选中URL路径，在右侧转发规则详情区域单击绑定。

基本信息 监听器管理 重定向配置 监控 安全组

温馨提示：当您配置了自定义重定向策略，原转发规则进行修改后，重定向策略会默认解除，需要重新配置。查看

HTTP/HTTPS监听器

新建

Listener1(HTTP:80)

www.example.com

/image/

转发规则详情 展开

已绑定后端服务

绑定 修改端口 修改权重 解绑

2. 在绑定后端服务对话框中，选择绑定实例类型为“云服务器”，再选择与 CLB 实例同地域下的云服务器实例 rs-1 和 rs-2，设置云服务器端口均为“80”，云服务器权重均为默认值“10”，然后单击确认。

绑定后端服务

请选择实例

已选择 (2)

ID/实例名	端口	权重 ①
(rs-2)	80	- 10 +
(rs-1)	80	- 10 +

云服务器 弹性网卡 请输入默认端口

IP地址 按照IP地址搜索, 关键字

ID/实例名

(rs-2)

(rs-1)

每页显示行 10 / 1 页

支持按住 shift 键进行多选

确认 取消

3. 返回“转发规则详情”区域，可以查看绑定的云服务器和其健康检查状态，当端口健康状态为“健康”时表示云服务器可以正常处理负载均衡转发的请求。

说明：

一条转发规则（监听协议 + 端口 + 域名 + URL 路径）可以绑定同一台云服务器的多个端口。如用户在 rs-1 的 80 和 81 端口部署了一样的服务，则 CLB 支持示例中的转发规则同时绑定 rs-1 的 80 和 81 端口，两个端口都会接收到 CLB 转发的请求。

## 步骤4：配置安全组

创建完负载均衡后，您可以配置负载均衡的安全组来隔离公网流量，详情请参考 [配置安全组](#)。安全组配置完成后，您可以选择开启或关闭安全组默认放通，不同选择配置如下所示。

### 方法一：开启安全组默认放通

具体操作请参考 [配置安全组默认放通](#)。

### 方法二：在 CVM 安全组上放通客户端 IP

具体操作请参考 [配置安全组默认放通](#)。

## 步骤5：添加域名解析

域名注册成功后，您可为域名添加域名解析，以便通过域名访问网站。

1. 登录 [云解析 DNS 控制台](#)，在域名解析列表页面，单击目标域名右侧操作列的解析，本文档以 example.com 域名为例。实际操作以域名化负载均衡实例的域名为准，具体可以在实例列表页查询，域名后缀为-tencentclb.cloud。

2. 在记录管理页签，单击添加记录。

3. 在添加记录区域，填写以下参数：

3.1 按需填写主机记录，主机记录就是域名前缀，详情请参见 [子域名说明](#) 和 [泛解析说明](#)，常见用法有：

- www**: 解析后的域名为 www.example.com。
- @**: 直接解析主域名 example.com。
- \***: 泛解析，匹配其他所有域名 \*.example.com。

3.2 选择记录类型，推荐选择 CNAME。

提示  
将域名指向云服务器,请选择「A」;  
将域名指向另一个域名,请选择「CNAME」;  
建立邮箱请选择「MX」,根据邮箱服务商提供的MX记录填写。

**A** 用来指定域名的IPv4地址(如:8.8.8.8),如果需要将域名指向一个IP地址,就需要添加A记录。 **点击自动填充**

**CNAME** 如果需要将域名指向另一个域名,再由另一个域名提供IP地址,就需要添加CNAME记录。

**MX** 如果需要设置邮箱,让邮箱能收到邮件,就需要添加MX记录。

**TXT** 在这里可以填写任何东西,长度限制255。绝大多数的TXT记录是用来做SPF记录(反垃圾邮件)。

**NS** 域名服务器记录,如果需要把子域名交给其他DNS服务商解析,就需要添加NS记录。

**AAAA** 用来指定主机名(或域名)对应的IPv6地址(例如:ff06:0:0:0:0:c3)记录。

**SRV** 记录了多台计算机提供了哪个服务,格式为:服务的名字、点、协议的类型,例如:\_xmpp-server\_tcp。

**显性URL** 从一个地址301重定向到另一个地址的时候,就需要添加显性URL记录(注:DNSPod目前只支持301重定向)。

**隐性URL** 类似于显性URL,区别在于隐性URL不会改变地址栏中的域名。

**3.3 线路类型:选择“默认”类型,否则会导致部分用户无法解析。例如,您需要将联通用户指向 2.com,所有非联通用户都指向 1.com。您可以通过添加线路类型为默认、记录值为 1.com 和线路类型为联通、记录值为 2.com 的两条 CNAME 记录来实现。**

**3.4 填写记录值, CNAME 记录值可填写为负载均衡提供给您的域名。**

**3.5 其余值可以保持默认值,操作完成后,单击保存。**

提示  
填写一个域名,如:cloud.tencent.com

**4. 添加记录完成后,可以在记录管理页签的记录列表查看刚才添加的记录。**

主机记录	记录类型	线路类型	记录值	MX优先级	TTL (秒)	最后操作时间	操作
www	CNAME	默认	1b16c9-0.ap-guangzhc	-	600	2019-02-18 18:36:50	修改 停用 删除
www	A	默认	负载均衡	详情	-	600	2019-02-18 18:36:50

## 步骤6: 验证负载均衡服务

在添加完解析记录十分钟后,在浏览器中输入绑定后的 CNAME 域名(如本例中的 `www.example.com/image/`),若正常显示页面,则表示负载均衡生效。

## 配置重定向功能(可选)

- 重定向配置分为手动重定向和自动重定向，详情请参见 [重定向配置](#)：  
自动重定向（强制 HTTPS）：PC 端、手机浏览器等以 HTTP 请求访问 Web 服务，在 CLB 代理后，返回 HTTPS 响应。默认强制浏览器以 HTTPS 访问网页。
- 手动重定向：当出现 Web 业务需要临时下线（如电商售罄、页面维护和更新升级时）需要重定向能力。如果不做重定向，用户的收藏和搜索引擎数据库中的旧地址只能让访客得到一个404或503错误信息页面，降低了用户体验度，导致访问流量白白流失，且该页面积累的搜索引擎评分也会无效。

## 相关操作

- [Linux \( CentOS \) 下部署 Java Web](#)
- [Windows 下安装备置 PHP](#)

# 负载均衡快速入门

最近更新时间：2025-02-05 15:52:41

腾讯云负载均衡支持 TCP、UDP、TCP SSL、HTTP 和 HTTPS 协议，提供基于域名和 URL 路径的灵活转发能力。本文将引导您如何快速创建一个负载均衡（CLB）实例，并将来自客户端的访问请求转发到两台后端云服务器（CVM）上。

## 前提条件

1. 已创建两个云服务器实例（本文以 `rs-1` 和 `rs-2` 实例为例）。有关如何创建云服务器实例，请参考 [购买并启动云服务器实例](#)。
2. 已在云服务器上搭建后端服务。本文以 HTTP 转发为例，已在两台云服务器上部署 Nginx 服务器，并在 `rs-1` 和 `rs-2` 分别返回一个带有“Hello nginx! This is rs-1!”和“Hello nginx! This is rs-2!”的 HTML，详情请参考 [CentOS 下部署 Nginx](#)。

### ⚠ 注意：

- 本文以标准账户类型（原“带宽上移账户”）为例。传统账户类型（原“带宽非上移账户”）的 CVM 必须购买公网带宽，因为当前的带宽属性在 CVM 上，而非 CLB 上。若您无法确定账户类型，请参见 [判断账户类型](#)。当前传统型负载均衡购买入口已关闭，如存量实例需要升级，请参考 [传统型负载均衡升级](#)。
- 示例中后端服务器部署的服务返回值不同，实际情况中，为保持所有用户均有一致体验，后端服务器上一般是部署完全相同的服务。

3. 域名注册是在互联网上建立服务的基础。

- 如果您已经在其他注册商拥有了自己的域名，您可以将域名转入腾讯云域名服务，详情请参见 [域名转入腾讯云](#)。
- 如果您还没有域名，您需要进行域名注册，详情请参见 [域名注册](#)。

## 步骤一：购买负载均衡实例

负载均衡实例成功购买后，系统将自动分配一个 VIP，该 VIP 为负载均衡向客户端提供服务的 IP 地址。

1. 登录腾讯云 [负载均衡服务购买页](#)。
2. 在负载均衡 CLB 购买页面，地域选择与云服务器相同的地域，实例类型选择负载均衡，网络类型选择公网。更多参数详情请参考 [产品属性选择](#)。

### ⓘ 说明：

目前仅广州、上海、南京、济南、杭州、福州、北京、石家庄、武汉、长沙、成都、重庆地域支持静态单线 IP 线路类型，其他地域支持情况请以控制台页面为准。如需体验，请联系商务经理申请。申请通过后，即可在购买页选择中国移动、中国联通或中国电信的运营商类型。

3. 单击立即购买，完成付款。
4. 返回实例管理页面，选择对应的地域即可看到新建的实例。

## 步骤二：配置负载均衡监听器

负载均衡监听器通过指定协议及端口来负责实际转发。本文以负载均衡转发客户端的 HTTP 请求配置为例。

### 配置 HTTP 监听协议和端口

当客户端发起请求时，负载均衡会根据监听的前端协议与端口接收请求并向后端服务器转发请求。

1. 登录 [负载均衡控制台](#)。
2. 在“实例管理”页面中，找到目标负载均衡实例，单击配置监听器。
3. 在“监听器管理”页签中的“HTTP/HTTPS 监听器”区域下，单击新建。



The screenshot shows the Listener Management interface. At the top, there are tabs: 基本信息, 监听器管理 (Listener Management, highlighted with a red box), 重定向配置, 监控, and 安全组. Below the tabs, a yellow box contains a warning message: “温馨提示：当您配置了自定义重定向策略，原转发规则进行修改后，重定向策略会默认解除，需要重新配置。查看” (Tip: When you configure a custom redirection strategy, if the original forwarding rule is modified, the redirection strategy will be automatically removed, and you need to reconfigure it. View). The main area is titled “HTTP/HTTPS 监听器” and features a “新建” (New) button, which is also highlighted with a red box.

4. 在“创建监听器”对话框中，配置以下内容，配置完成后，单击提交。
  - 监听器名称，名称可输入60个字符，允许英文字母、数字、中文字符、“-”、“\_”、“.”。
  - 监听协议端口。例如 `HTTP: 80`。

### 配置监听器的转发规则

当客户端请求时，负载均衡会根据配置监听器的转发规则进行请求转发。

1. 在“监听器管理”页签中，选中刚才新建的监听器，单击 `+ 添加规则`。



The screenshot shows the Listener Management interface with Listener1 (HTTP:80) selected. A dialog box is open with the title “添加规则” (Add Rule). Inside the dialog, there is a “+” button and a “添加规则” (Add Rule) button, both of which are highlighted with red boxes.

2. 在“创建转发规则”对话框的“基本配置”页签中，配置域名、URL路径和均衡方式。配置完成后，单击下一步。

- **域名：**您的后端服务所使用的域名，本例使用 `www.example.com`。详情请参见 [配置负载均衡的转发域名](#)。
- **默认域名：**当客户端请求没有匹配本监听器的任何域名时，CLB 会将请求转发给默认域名（`default server`），每个监听器只能配置一个默认域名。若该监听器没有配置默认域名时，CLB 会将请求转发给第一个域名。本例不配置。
- **URL 路径：**您的后端服务的访问路径，本例使用 `/image/`。
- **均衡方式选择“加权轮询”。**详情请参见 [均衡方式](#)。



3. 在“健康检查”页签，开启健康检查，检查域名和检查路径使用默认的转发域名和转发路径，完成后单击下一步。

## 创建转发规则



4. 在“会话保持”页签，关闭会话保持，单击提交。

有关负载均衡监听器的更多内容，请参考 [负载均衡监听器概述](#)。

**说明:**

- 转发规则：一个监听器可配置多个域名，一个域名可配置多条 URL 路径，选中监听器或域名，单击 +，即可创建新的规则。
- 会话保持：若关闭会话保持功能，选择轮询的方式进行调度，则请求会被依次分配到不同后端服务器上；若开启会话保持功能，或关闭会话保持功能但选择 ip\_hash 的调度方式，则请求会被持续分配到同一台后端服务器上。

## 为监听器绑定后端云服务器

当客户端请求时，负载均衡将请求转发到监听器绑定的云服务器上进行处理。

1. 在“监听器管理”页面，单击+展开刚才创建的监听器，选中 URL 路径，在右侧“转发规则详情”区域单击绑定。

2. 在“绑定后端服务”对话框中, 选择绑定实例类型为“云服务器”, 再选择与 CLB 实例同地域下的云服务器实例 `rs-1` 和 `rs-2`, 设置云服务器端口均为“80”, 云服务器权重均为默认值“10”, 然后单击确认。

ID/Instance Name	Port	Weight
rs-1 (未命名)	1-65535	10
rs-2 (未命名)	1-65535	10

3. 返回“转发规则详情”区域, 可以查看绑定的云服务器和其健康检查状态, 当端口健康状态为“健康”时表示云服务器可以正常处理负载均衡转发的请求。

**说明：**

一条转发规则（监听协议 + 端口 + 域名 + URL 路径）可以绑定同一台云服务器的多个端口。如用户在 rs-1 的 80 和 81 端口部署了一样的服务，则 CLB 支持示例中的转发规则同时绑定 rs-1 的 80 和 81 端口，两个端口都会接收到 CLB 转发的请求。

## 步骤三：配置安全组

创建完负载均衡后，您可以配置负载均衡的安全组来隔离公网流量，详情请参考 [配置安全组](#)。

安全组配置完成后，您可以选择开启或关闭安全组默认放通，不同选择配置如下所示。

### 方法一：开启安全组默认放通

具体操作请参考 [配置安全组默认放通](#)。

### 方法二：在 CVM 安全组上放通客户端 IP

具体操作请参考 [配置安全组默认放通](#)。

## 步骤四：验证负载均衡服务

配置完成负载均衡后，可以验证负载均衡是否生效，即验证通过一个负载均衡实例下不同的 域名+URL 访问不同的后端云服务器，也可验证内容路由（content-based routing）的功能。

### 方法一：配置 hosts 将域名指向 CLB

1. 在 Windows 系统中，进入 `C:\Windows\System32\drivers\etc` 目录，修改 hosts 文件，把域名映射到 CLB 实例的 VIP 上。

```
# localhost name resolution is handled within DNS itself.
# 127.0.0.1      localhost
# ::1            localhost
www.example.com
```

2. 为了验证 hosts 是否配置成功，可以运行 cmd 命令行工具，用 `ping` 命令探测一下该域名是否成功绑定了 VIP，如有数据包，则证明绑定成功。

```
C:\Users\w      >ping www.example.com

正在 Ping www.example.com [1] 具有 32 字节的数据:
来自 1 [REDACTED] 的回复: 字节=32 时间=7ms TTL=46

1  [REDACTED] 的 Ping 统计信息:
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
最短 = 7ms, 最长 = 7ms, 平均 = 7ms
```

3. 在浏览器中输入访问路径 `http://www.example.com/image/`，测试负载均衡服务。若如下图所示，则表示本次请求被 CLB 转发到了 rs-1 这台 CVM 上，CVM 正常处理请求并返回页面。



4. 此监听器的均衡方式是“加权轮询”，且两台 CVM 的权重都是“10”。刷新浏览器，再次发送请求，若如下图所示，则表示本次请求被 CLB 转发到了 rs-2 这台 CVM 上。



#### ⚠ 注意：

`image/` 后 `/` 必须保留，代表 `image` 是默认的目录，而不是名为 `image` 的文件。

## 方法二：配置云解析 DNS 将域名指向 CLB

1. 打开 [腾讯云域名注册页面](#) 进行域名查询和注册。本例以 `example.com` 为例，详情请参考 [域名注册](#)。
2. 登录 [云解析 DNS 控制台](#)，在“域名解析列表”页面，单击目标域名右侧“操作”列的解析。
3. 在“记录管理”页签，单击添加记录为域名添加 A 记录，在添加记录区域，填写以下参数：
  - 主机记录：指域名前缀。以泛解析为例，它可以匹配其他所有域名前缀，设置为 `*`。
  - 记录类型： `A记录`。
  - 线路类型： `默认`。
  - 记录值：单击[关联云资源](#)，在“关联云资源”对话框勾选刚才创建的 CLB 实例。
  - TTL：设置为默认值“600s”。
4. 添加完成后，单击保存。
5. 在添加完解析记录十分钟后，在浏览器中输入绑定后的 CNAME 域名（如本例中的 `www.example.com/image/`），若正常显示页面，则表示负载均衡生效。

## 配置重定向功能（可选）

重定向配置分为手动重定向和自动重定向，详情请参见 [重定向配置](#)：

- **自动重定向（强制 HTTPS）**：PC 端、手机浏览器等以 HTTP 请求访问 Web 服务，在 CLB 代理后，返回 HTTPS 响应。默认强制浏览器以 HTTPS 访问网页。

- **手动重定向：**当出现 Web 业务需要临时下线（如电商售罄、页面维护和更新升级时）需要重定向能力。如果不做重定向，用户的收藏和搜索引擎数据库中的旧地址只能让访客得到一个404或503错误信息页面降低了用户体验度，导致访问流量白白流失，且该页面积累的搜索引擎评分也会无效。

## 相关操作

- [Linux \( CentOS \) 下部署 Java Web](#)
- [Windows 下安装配置 PHP](#)

# IPv6 负载均衡快速入门

最近更新时间：2025-10-11 10:57:51

腾讯云负载均衡支持 IPv4、IPv6 和 IPv6 NAT64 三个 IP 版本，IPv6 负载均衡支持 TCP/UDP/TCP SSL/HTTP/HTTPS 协议，提供基于域名和 URL 路径的灵活转发能力。本文将引导您如何快速使用 IPv6 负载均衡。

## 说明

IPv6 负载均衡处于内测阶段，如需使用，请提交 [工单申请](#)。

## 前提条件

1. 负载均衡只负责转发流量，不具备处理请求的能力。因此，您需要首先搭建处理用户请求的云服务器实例，并完成云服务器的 IPv6 配置。有关如何创建云服务器实例并启用 IPv6，请参见 [快速搭建 IPv6 私有网络](#)。
2. 本文以 HTTP 转发为例，且 Linux 云服务器上必须部署相应的 Web 服务器（如 Apache、Nginx、IIS 等），同时 Web 服务使用的端口需要监听 IPv6。

## 使用说明

- 目前仅支持如下地域开通 IPv6 负载均衡：广州、深圳金融、上海、上海金融、南京、北京、北京金融、成都、重庆、中国香港、新加坡、弗吉尼亚、圣保罗。其中，针对地域为深圳金融、上海金融的金融行业监管要求定制的合规专区，需提交 [工单申请](#) 使用专区。
- IPv6 负载均衡不支持传统型负载均衡。
- IPv6 负载均衡支持获取客户端 IPv6 源地址。四层 IPv6 负载均衡支持直接获取客户端 IPv6 源地址，七层 IPv6 负载均衡支持通过 HTTP 的 X-Forwarded-For 头域获取客户端 IPv6 源地址。
- 当前 IPv6 负载均衡是纯公网负载均衡，相同 VPC 的客户端无法访问该 IPv6 负载均衡。
- 互联网 IPv6 网络大环境还处于建设初期，如出现线路访问不通的情况，请 [提交工单](#) 反馈。

## 步骤1：搭建云服务器并配置 IPv6

1. 进入 [云服务器控制台](#)，登录云服务器完成 IPv6 的基础配置，详细操作请参见 [快速搭建 IPv6 私有网络](#)。
2. 在云服务器中，依次执行如下命令，部署并重启 Nginx 服务。

```
yum install nginx
service nginx restart
```

3. 查看部署在云服务器上的 Nginx 服务是否已经监听 IPv6。

3.1 执行如下命令进行查看。

```
netstat -tupln
```

```
[root@VM_0_14_centos ~]# netstat -tupln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 0.0.0.0:80              0.0.0.0:*
tcp      0      0 0.0.0.0:22              0.0.0.0:*
tcp6     0      0 :::80                  :::*
tcp6     0      0 0.0.0.0:68              0.0.0.0:-
tcp6     0      0 10.24.0.14:123         0.0.0.0:*
tcp6     0      0 127.0.0.1:123         0.0.0.0:*
tcp6     0      0 0.0.0.0:56713         0.0.0.0:*
udp      0      0 fe80::5054:ff:fe3d::546  :::*
udp6     0      0 2402:4e00:1400:1217:123  :::*
udp6     0      0 fe80::5054:ff:fe3d::123  :::*
udp6     0      0 ::1:123                :::*
```

3.2 执行如下命令，打开 Nginx 配置文件进行查看。

```
vim  /etc/nginx/nginx.conf
```

```
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    listen      80 default server;
    listen      [::]:80 default_server;
    server_name ;
    root       /usr/share/nginx/html;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}

# Settings for a TLS enabled server.

server {
    listen      443 ssl http2 default server;
    listen      [::]:443 ssl http2 default_server;
    server_name ;
    root       /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    ssl_session_cache shared:SSL:1m;
}
```

## 步骤2：创建 IPv6 负载均衡实例

1. 登录腾讯云官网，进入 [负载均衡购买页](#)。
2. 请正确选择如下参数：
  - 计费模式：仅支持按量计费。
  - 地域：选择目标地域。
  - IP 版本：IPv6。
  - 运营商类型：BGP（多线）。
  - 网络：请务必选择已获取 IPv6 CIDR 的私有网络和子网。
3. 在购买页选择各项配置后，单击立即购买。

4. 在“CLB 实例列表”页，选择对应的地域即可看到新建的实例。

ID	ID名称	监控	状态	VIP	网络类型	所属网络	健康状态	创建时间	操作
1	1	已	正常	2	公网	3	健康检查未配置(配置)	2019-11-25 17:32:26	配置监听器 更多

## 步骤3：创建 IPv6 负载均衡监听器

### 配置 HTTP 监听协议和端口

1. 登录 [负载均衡控制台](#)。
2. 在CLB 实例列表中，找到已创建的负载均衡实例，单击实例 ID，进入负载均衡详情页。
3. 在基本信息模块，可以单击名称后的修改图标修改实例名称。
4. 在监听器管理中的 HTTP/HTTPS 监听器下，单击新建，新建负载均衡监听器。



监听器管理

HTTP/HTTPS监听器

新建

您还未创建监听器，点击[开始创建](#)

点击左侧节点查看详情

5. 在弹出框中，配置以下内容：

- 名称自定义为“IPv6test”。
- 监听协议端口为 `HTTP: 80`。

6. 单击提交，创建负载均衡监听器。

### 配置监听器的转发规则

1. 在监听器管理中，选中刚才新建的监听器 `IPv6test`，单击右边的 `+`，开始添加规则。
  2. 在弹出框中，配置域名、URL 路径和均衡方式，单击下一步。
- 域名：您的后端服务所使用的域名，本例使用 `www.xxxxxxxxxxxtest.com`。域名支持通配符，详情请参见 [七层转发域名和 URL 规则说明](#)。
  - URL 路径：您的后端服务的访问路径，本例使用 `/`。
  - 均衡方式选择加权轮询。

## 创建转发规则



1 基本配置 > 2 健康检查 > 3 会话保持

域名 ⓘ

[新增域名](#)

默认域名

当客户端请求没有匹配本监听器的任何域名时, CLB会将请求转发给默认域名 (Default Server), 每个监听器只能配置且必须配置一个默认域名, [详情](#)

URL路径 ⓘ

均衡方式 ⓘ

WRR 根据新建连接数来调度, 权重越高的后端服务器被轮询到的概率越高

获取客户端IP  已启用

Gzip压缩  已启用 ⓘ

[关闭](#)

[下一步](#)

3. 配置健康检查: 开启健康检查, 检查域名使用默认的转发域名和转发路径, 单击下一步。

健康检查源地址: fd76:3655:6c63::/64

## 创建转发规则



1 基本配置 > 2 健康检查 > 3 会话保持

健康检查



帮助您自动检查并移除异常的后端服务器。

健康探测源IP fd76:3655:6c63::/64 网段

检查方式  TCP  HTTP检查域名 ⓘ 不填则默认为转发域名检查路径 ⓘ 后端服务器的根目录 /[显示高级选项](#) ▾

上一步

下一步

4. 会话保持：开启会话保持并配置保持时间，单击提交。

## 创建转发规则



1 基本配置 > 2 健康检查 > 3 会话保持

会话保持 ⓘ 保持时间 ⓘ 30秒 86400秒 - 30 + 秒

基于cookie植入的会话保持

上一步

提交

有关负载均衡监听器的更多内容，请参见 [负载均衡监听器概述](#)。

**说明：**

- 一个监听器（即监听协议：端口）可以配置多个域名，一个域名下可以配置多条 URL 路径，选中监听器或域名，单击+即可创建新的规则。
- 会话保持：如果用户关闭会话保持功能，选择轮询的方式进行调度，则请求依次分配到不同后端服务器上；如果用户开启会话保持功能，或关闭会话保持功能但选择 ip\_hash 的调度方式，则请求持续分配到同一台后端服务器上去。

## 绑定云服务器

**说明：**

绑定云服务器前，请确定该云服务器已获取 IPv6 地址。

- 在监听器管理页面，选中并展开刚才创建的监听器，选中域名、选中 URL 路径，在右侧即可看到该 URL 路径绑定的云服务器 IPv6 信息，单击绑定。
- 在弹框中，选择云服务器，并设置云服务器的 Nginx 服务默认端口为80，设置权重（默认值10），单击确定。



### 3. 成功绑定云服务器后：

- 请确认端口状态是否为健康，如果为健康，请进行 [步骤4：测试 IPv6 负载均衡](#)。

HTTP/HTTPS监听器

新建

— IPv6test(HTTP:80)

— www.qcloudipv6test.com

— /image/

转发规则详情 展开

已绑定后端服务

绑定	修改端口	修改权重	解绑			
<input type="checkbox"/>	CVM ID/名称	端口状态	IP地址	端口	权重	操作
<input type="checkbox"/>	2	健康	2	80	10	解绑
<input type="checkbox"/>	0	健康	2	80	10	解绑
<input type="checkbox"/>	e	健康	0	80	10	解绑

- 如果端口状态为异常，请确定监听器是否绑定了正确的云服务器的 Nginx 服务端口，同时登录云服务器检查端口已经正常监听 IPv6，可参见 [步骤1中的第3步](#) 进行查看。

HTTP/HTTPS监听器

新建

— IPv6test(HTTP:80)

— www.qcloudipv6test.com

— /image/

转发规则详情 展开

已绑定后端服务

绑定	修改端口	修改权重	解绑			
<input type="checkbox"/>	CVM ID/名称	端口状态	IP地址	端口	权重	操作
<input type="checkbox"/>	2	异常	2	88	10	解绑
<input type="checkbox"/>	8	异常	2	88	10	解绑
<input type="checkbox"/>	b	异常	8	88	10	解绑

## 步骤4：测试 IPv6 负载均衡

配置完成 IPv6 负载均衡后，可以验证该架构是否生效，即验证通过一个 CLB 实例下不同的域名 + URL 访问不同的后端云服务器，验证内容路由 (content-based routing) 的功能是否可用。

使用具有 IPv6 公网能力的客户端，访问域名或者负载均衡的 IPv6 地址，如果能够正常访问云服务器的 Web 服务，则表明 IPv6 负载均衡工作正常，示例步骤如下：

- 打开 [腾讯云域名注册页面](#) 进行域名查询和注册。本例以 `xxxxxxxxxtest.com` 为例，详情请参考 [域名注册](#)。
- 登录 [云解析 DNS 控制台](#)，单击您所购买的域名，在“记录管理”页面单击添加记录，为域名添加 AAAA 记录，输入如下内容并保存：
  - 主机记录：即域名前缀，本例设为 `www`。

- 记录类型：AAAA记录。
- 线路类型：默认。
- 记录值：填写负载均衡的 IPv6 地址。
- TTL：设置为默认值“600s”。

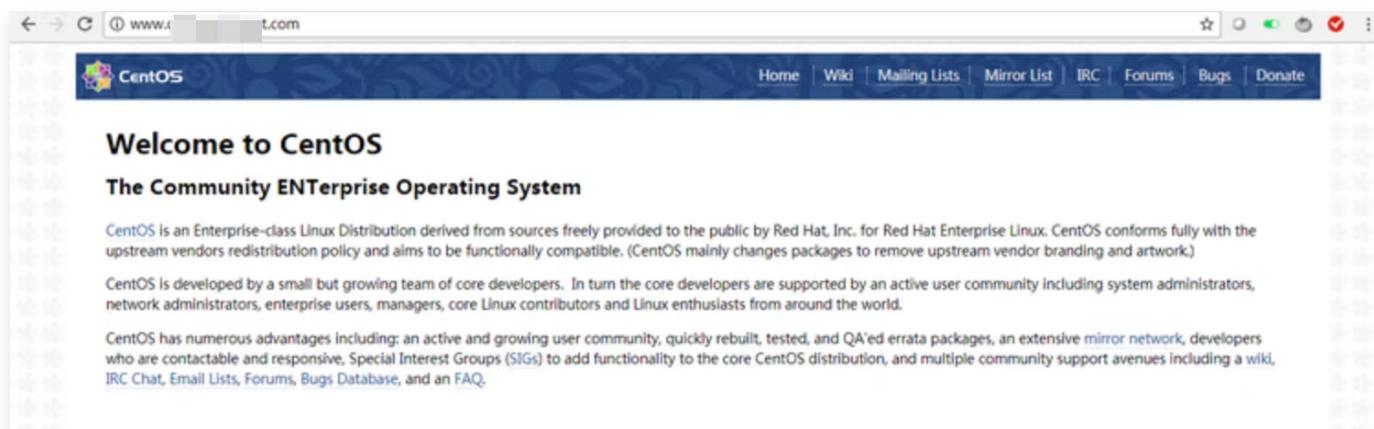
### 3. 添加域名解析后，通过 Ping 域名进行验证，如下图：

```
C:\Users\andyzhqin>ping -6 www.172.16.1.100.t.com

正在 Ping www.172.16.1.100.t.com [2402: :2482] 具有 32
字节的数据:
来自 2402:          2482 的回复: 时间=44ms
来自 2402:          2482 的回复: 时间=33ms
来自 2402:          2482 的回复: 时间=32ms

2402:          :2482 的 Ping 统计信息:
数据包: 已发送 = 3, 已接收 = 3, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
最短 = 32ms, 最长 = 44ms, 平均 = 36ms
```

### 4. 再通过浏览器访问域名来验证，如下图：



# CentOS 下部署 Nginx

最近更新时间：2024-12-02 16:45:22

本文将为您详细介绍如何在 CentOS 系统下部署 Nginx 项目，适用于刚开始使用腾讯云的个人用户。

## 软件版本

本文在示例步骤中的软件版本如下，在实际操作时，请您以实际软件版本为准。

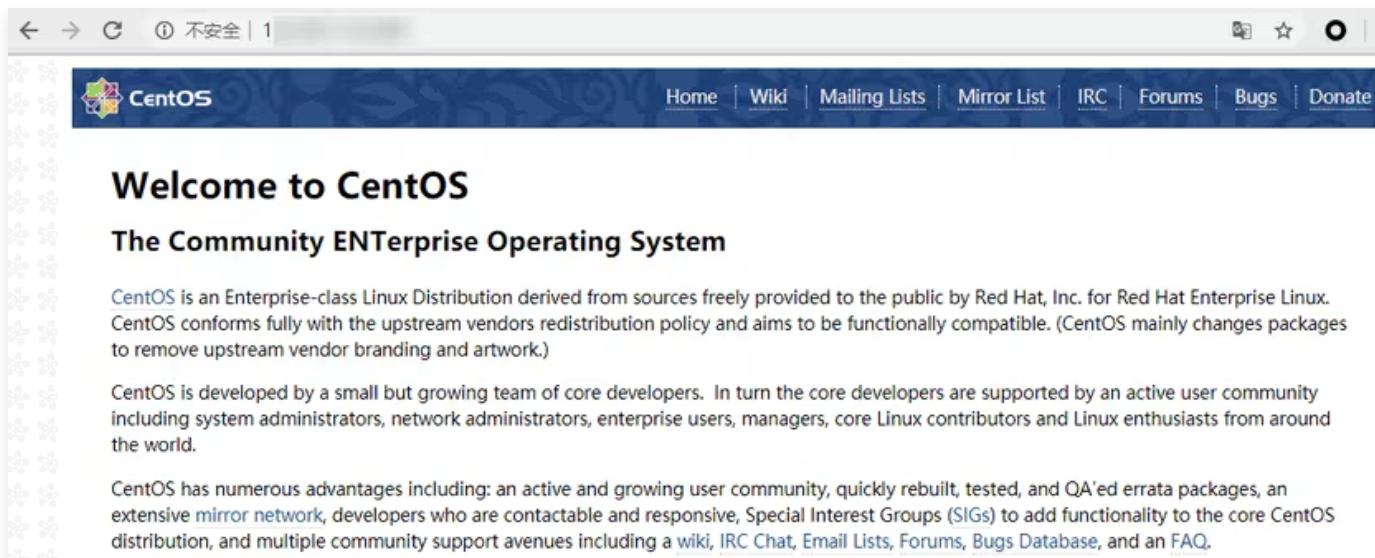
- 操作系统：CentOS 7.5
- Nginx 版本：Nginx 1.16.1

## 安装 Nginx

1. 购买完成后，在云服务器的详情页面，单击登录，可以直接登录云服务器，输入自己的用户名密码后，开始搭建 Nginx 环境。有关如何创建云服务器实例，请参见 [创建云服务器实例](#)。

```
# 安装 Nginx:  
yum -y install nginx  
# 查看 Nginx 版本  
nginx -v  
# 查看 Nginx 安装目录  
rpm -ql nginx  
# 启动 Nginx  
service nginx start
```

2. 访问该云服务器的公网 IP 地址，出现如下页面则表示 Nginx 部署完成：



The screenshot shows a web browser displaying the official CentOS website. The page features a dark blue header with the CentOS logo and navigation links for Home, Wiki, Mailing Lists, Mirror List, IRC, Forums, Bugs, and Donate. The main content area is titled "Welcome to CentOS" and "The Community ENTerprise Operating System". It includes a brief description of CentOS as an Enterprise-class Linux distribution and its development by a community of core developers and users. The page also highlights CentOS's advantages, such as an active user community, quick rebuilds, and extensive support resources.

3. Nginx 的默认根目录 root 是 /usr/share/nginx/html，直接修改 html 下的 index.html 静态页面，用来标识这个页面的特殊性，相关操作如下：

3.1 执行如下命令，进入html 下的 index.html 静态页面：

```
vim /usr/share/nginx/html/index.html
```

3.2 按 “i” 进入编辑模式，请在 <body></body> 标签内输入：

```
# 建议直接在<body>下方输入
Hello nginx , This is rs-1!
URL is index.html
```

```
<body>
Hello nginx , This is rs-1!
URL is index.html
```

3.3 按 “Esc” ，输入 :wq 保存编辑并退出。

4. 负载均衡（原“应用型负载均衡”）可以根据后端服务器的路径来进行请求转发，在 /image 路径下部署静态页面，相关操作如下：

4.1 依次执行如下命令，新建目录 image 并进入该目录：

```
mkdir /usr/share/nginx/html/image
cd /usr/share/nginx/html/image
```

4.2 执行如下命令，在 image 目录下创建 index.html 静态页面：

```
vim index.html
```

4.3 按 “i” 进入编辑模式，在页面中输入：

```
Hello nginx , This is rs-1!
URL is image/index.html
```

4.4 按 “Esc” ，输入 :wq 保存编辑并退出。

#### ⚠ 注意：

Nginx 的默认端口是 80，如果想修改端口请修改配置文件并重启 Nginx。

## 验证 Nginx 服务

访问云服务器的公网 IP + 路径，如果可以显示出已部署好的静态页面，则证明 Nginx 部署成功。

- rs-1 的 index.html 页面：



- rs-1 的 /image/index.html 页面：



# CentOS 下部署 Java Web

最近更新时间：2024-11-22 16:19:44

本文将为您详细介绍如何在 CentOS 系统下部署 Java Web 项目，适用于刚开始使用腾讯云的个人用户。

## 软件版本

本文在示例步骤中的软件版本如下，在实际操作时，请您以实际软件版本为准。

- 操作系统：CentOS 7.5
- Tomcat 版本：apache-tomcat-8.5.39
- JDK 版本：JDK 1.8.0\_201

## 安装JDK

购买负载均衡服务后，在云服务器的详情页面，单击登录，可以直接登录云服务器，输入自己的用户名密码后，开始搭建 Java Web 环境。有关如何创建云服务器实例，请参见 [云服务器-创建实例](#)。

### 下载 JDK

输入如下命令：

```
mkdir /usr/java    # 创建 java 文件夹
cd /usr/java      # 进入 java 文件夹
```

# 上传 JDK 安装包（推荐）

推荐您使用 [WinSCP](#) 或其他工具将 JDK 安装包上传到上述 java 文件夹下，然后解压安装包。

或者

# 直接使用命令（推荐您使用上传 JDK 安装包的方法）： wget 下载链接，下载得到的压缩包无法解压，这是因为直接下载的压缩包默认没有接受 Oracle BSD 许可；每个人的 cookie 不一样，请前往<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>页面同意许可协议并获取带有自己 cookie 的下载链接。

```
wget --no-check-certificate --no-cookies --
header "Cookie: oraclelicense=accept-securebackup-
cookie" https://download.oracle.com/otn-pub/java/jdk/8u201-
b09/42970487e3af4f****bca3f542482c60/jdk-8u201-linux-x64.tar.gz
```

# 解压

```
chmod +x jdk-8u201-linux-x64.tar.gz
```

```
tar -xzvf jdk-8u201-linux-x64.tar.gz
```

## 设置环境变量

1. 打开 `/etc/profile` 文件。

```
vi /etc/profile
```

2. 按下 `i` 键进入编辑模式，在该文件中添加如下信息。

```
# set java environment
export JAVA_HOME=/usr/java/jdk1.8.0_201
export
CLASSPATH=$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/rt.jar:$JAVA_HOME/li
b
export PATH=$JAVA_HOME/bin:$PATH
```

3. 按下 `Esc` 键退出编辑模式，输入 `:wq` 保存并关闭文件。

4. 加载环境变量。

```
source /etc/profile
```

## 查看 JDK 是否安装成功

运行 `java -version` 命令，显示 JDK 版本信息时，表示 JDK 已经安装成功。

```
[root@emma ~]# java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
```

## 安装 Tomcat

### 下载 Tomcat

输入如下命令：

```
# 镜像地址会改变，Tomcat 版本也会不断升级。如果下载链接失效，请您到 [Tomcat 官网]
(https://tomcat.apache.org/download-80.cgi) 选择合适的安装包地址。
wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-8/v8.5.39/bin/apache-tomcat-8.5.39.tar.gz
```

```
tar -xzvf apache-tomcat-8.5.39.tar.gz
mv apache-tomcat-8.5.39 /usr/local/tomcat/
```

在 /usr/local/tomcat/ 目录中包含如下文件：

- **bin**: 脚本文件，包含启动和关闭 Tomcat 服务脚本。
- **conf**: 各种全局配置文件，其中最重要的是 **server.xml** 和 **web.xml**。
- **webapps**: Tomcat 的主要 Web 发布目录，默认情况下把 Web 应用文件放于此目录。
- **logs**: 存放 Tomcat 执行时的日志文件。

#### ⚠ 注意：

如果下载链接失效，请替换为 [Tomcat 官网](#) 的最新下载链接。

## 添加用户

```
# 创建一般用户 www 来运行 Tomcat
useradd www

# 创建网站根目录
mkdir -p /data/wwwroot/default
# 将需要部署的 Java Web 项目文件 WAR 包上传到网站根目录下，然后将网站根目录下文件权限改为 www。本示例将直接在网站根目录下新建一个 Tomcat 测试页面：
echo Hello Tomcat! > /data/wwwroot/default/index.jsp
chown -R www.www /data/wwwroot
```

## 设置 JVM 内存参数

1. 创建一个 /usr/local/tomcat/bin/setenv.sh 脚本文件。

```
vi /usr/local/tomcat/bin/setenv.sh
```

2. 按下 i 键进入编辑模式，添加如下内容。

```
JAVA_OPTS='-Djava.security.egd=file:/dev/.urandom -server -Xms256m -
-Xmx496m -Dfile.encoding=UTF-8'
```

3. 按 Esc 键退出编辑模式，输入 :wq 保存并退出编辑。

## 配置 server.xml

1. 切换到 /usr/local/tomcat/conf/ 目录。

```
cd /usr/local/tomcat/conf/
```

## 2. 备份 server.xml 文件。

```
mv server.xml server_default.xml
```

## 3. 创建一个新的 server.xml 文件。

```
vi server.xml
```

## 4. 按下 i 键进入编辑模式，添加如下内容。

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8006" shutdown="SHUTDOWN">
  <Listener
    className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
  <Listener
    className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"
  />
  <Listener
    className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"
  />
  <Listener className="org.apache.catalina.core.AprLifecycleListener"/>
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml"/>
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080"
      protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443"
      maxThreads="1000"
      minSpareThreads="20"
      acceptCount="1000"
```

```
maxHttpHeaderSize="65536"
debug="0"
disableUploadTimeout="true"
useBodyEncodingForURI="true"
enableLookups="false"
URIEncoding="UTF-8"/>
<Engine name="Catalina" defaultHost="localhost">
<Realm className="org.apache.catalina.realm.LockOutRealm">
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
</Realm>
<Host name="localhost" appBase="/data/wwwroot/default"
unpackWARs="true" autoDeploy="true">
<Context path="" docBase="/data/wwwroot/default" debug="0"
reloadable="false" crossContext="true"/>
<Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t
&quot;%r&quot; %s %b" />
</Host>
</Engine>
</Service>
</Server>
```

5. 按 Esc 键退出编辑模式，输入 `:wq` 保存并退出编辑。

## 启动 Tomcat

### 方法一

进入 Tomcat 服务器的 bin 目录，然后执行 `./startup.sh` 命令启动 Tomcat 服务器。

```
cd /usr/local/tomcat/bin
./startup.sh
```

运行结果如下：

```
[root@emma bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_201
Using CLASSPATH:        /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

## 方法二

1. 创建一个 `/etc/init.d/tomcat` 的脚本文件。

```
vi /etc/init.d/tomcat
```

2. 按下 `i` 键进入编辑模式，添加如下内容。

```
#!/bin/bash
#
# tomcat startup script for the Tomcat server
#
#
# chkconfig: 345 80 20
# description: start the tomcat deamon
#
# Source function library
. /etc/rc.d/init.d/functions

prog=tomcat
export JAVA_HOME=/usr/java/jdk1.8.0_201
export CATALANA_HOME=/usr/local/tomcat

case "$1" in
start)
    echo "Starting Tomcat..."
    $CATALANA_HOME/bin/startup.sh
    ;;
stop)
    echo "Stopping Tomcat..."
    $CATALANA_HOME/bin/shutdown.sh
    ;;
restart)
    echo "Stopping Tomcat..."
    $CATALANA_HOME/bin/shutdown.sh
    sleep 2
    echo
    echo "Starting Tomcat..."
```

```
$CATALANA_HOME/bin/startup.sh
;;
status)
if pgrep -f "catalina.start" > /dev/null; then
    echo "Tomcat is running."
else
    echo "Tomcat is not running."
fi
;;
*)

echo "Usage: $prog {start|stop|restart|status}"
;;
esac
exit 0
```

### 3. 增加可执行权限。

```
chmod +x /etc/init.d/tomcat
```

### 4. 设置自启动。

```
chkconfig --add tomcat
chkconfig tomcat on
```

### 5. 启动 Tomcat。

```
# 启动 Tomcat
service tomcat start
# 查看 Tomcat 运行状态
service tomcat status
# 关闭 Tomcat
service tomcat stop
```

运行结果如下：

```
[root@emma bin]# service tomcat start
Tomcat is already running (pid: 27293)
[root@emma bin]# service tomcat status
Tomcat is running with pid: 27293
[root@emma bin]# service tomcat stop
Stopping Tomcat
Using CATALINA_BASE:    /usr/local/tomcat
Using CATALINA_HOME:    /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:          /usr/java/jdk1.8.0_201
Using CLASSPATH:         /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
waiting for processes to exit
```

6. 若提示没有权限，请切换为 root 用户并修改权限。

```
cd /usr/local
chmod -R 777 tomcat
```

7. 在浏览器地址栏中输入 `http://公网IP:端口` (端口为 `server.xml` 中设置的 `connector port`) 进行访问。出现下图所示页面时表示安装成功。



## 配置安全组

如果访问不通，请检查安全组。如上示例中 `server.xml` 中的 `connector port` 是8080，因此需在对应的云服务器所绑定的安全组上放通 TCP:8080，详情请参见 [添加安全组规则](#)。