

# Jexus v7.1.x 用户手册

---

宇内流云 (j66x@163.com)

## 目 录

一、Jexus 概述 .....	2
二、Jexus 的安装与更新 .....	6
三、Jexus 的操作命令 .....	9
四、Jexus 的全局配置 .....	12
五、Jexus 的网站配置 .....	14
六、关于 .....	25

## 一、Jexus 概述

Jexus 是 Linux 平台上具有高安全性、高可靠性、高稳定性的高性能 WEB 服务器、反向代理网关服务器和自宿主 WEB 应用程序托管服务器。

Jexus 除了具备通用 WEB 服务器所必备的功能外，还直接支持 Asp.Net WEB 应用，支持 PHP，支持反向代理。自 6.0 版开始，Jexus 开创性地设计了“AppHost”功能，通过此功能，Jexus 可以为 Asp.Net Core、Tomcat、Go/Python Web 等“HTTP 自宿主 WEB 应用程序”提供一体化集成管理。

就中国而言，Jexus 是一款完全由中国人自主研发的国产服务器软件，支持国产操作系统和国产 CPU 芯片，真正做到了“安全、可靠、可信、可控”，具备我国党政机关和重要企事业单位信息化建设所需要的关键品质。

### **Jexus v7.x 有如下特点：**

#### **1、性能强劲：**

在操作系统支持方面，Jexus 的设计目标非常明确：只考虑支持 Linux 和国产操作系统。正因为目标明确，定位清晰，针对性极强，所以在设计过程中完全不需要考虑兼容其它操作系统而不得不编写

的大量冗余代码，也无需使用低性能的折衷方案，而是直接使用 Linux 的长项技术，比如 `epoll` 技术、`sendfile` 零拷贝技术等等，核心技术直通操作系统。在编程技术方面，Jexus 使用了 `c/c++` 与 `c#` 的混合编程技术，`.Net Native` 技术，从语言层面为 Jexus 提供了坚实的保证。

## 2、安全可靠：

安全性是 Jexus 有别于其它 Web 服务器（比如 `Nginx`、`Apache`、`IIS` 等）的一个重要特色。Jexus 初始开发于 2006 年，2008 年在 `www.linuxdot.net` 上公开发布 1.0 版，最初开发的目的是为某政府网站提供 `Asp.Net` 跨平台支持，因此，Jexus 从诞生开始，就已将其安全性作为开发的首要目标，不但在编码中努力避免漏洞和安全缺陷，而且还开创性地内置了“入侵检测和控制”技术，并在长期的运行实践中不断地完善其策略，优化其算法，为 Jexus 托管的网站提供了强有力的安全保护。

可靠性是 Jexus 另一个突出优点。可靠性主要表现在两个方面，一是稳定性，二是“可信可控”特性。在稳定性方面，由于 Jexus 采用了独特的混合编程技术，通过语言层面的扬长避短，在高性能与高可用性方面得到了平衡。另外，10 多年来，Jexus 在各种生产场景中摸爬滚打，在“实战”环境中不断地发现问题解决问题，不断地完善，保证了 Jexus 持续可靠的稳定性。在“可信可控”特质方面，Jexus 是我国唯一一款由中国人（宇内流云/刘冰先生）从 0 开始原生开发的国产 WEB 服务器，整个 Jexus 程序系统没有使用任何特殊的第三方组件，甚至连开源代码也基本上没有“参考”使用，是纯粹的“中国

血统”，从程序开发者身份到程序源代码、程序运行环境等方面确保了 Jexus “安全可靠”“可控可信”的“信创”品质。

### 3、功能强大：

首先，Jexus 是一款如同 Nginx、Apache 那样的“通用 web 服务器”，具有通用 web 服务器必要的功能，除此之外，Jexus 还具备如下功能特征：

（1）支持 ASP.NET。在 Linux 上支持 Asp.Net 运行，是 Jexus 的重要的特色功能之一（甚至一度被部分用户误认为 Jexus 就是一款单纯的 Asp.Net 服务器）。Jexus 直接支持 .NET Framework 3.5—4.8 范围内的所有版本的 Asp.Net Web 程序。Jexus7.0 以上版本已经集成.NET 运行环境，服务器无需另行安装 mono 之类的.NET 运行时。

（2）提供 Fast-CGI 协议支持。通过 Fast-CGI，Jexus 能支持包括 PHP 在内的所有拥有 Fast-CGI 服务接口的 WEB 应用；

（3）具有强劲的反向代理功能。Jexus 支持全站反向代理、部分路径反向代理，支持多目标服务器反向代理（负载均衡），支持 TCP “穿透”技术。Jexus 的反向代理配置简洁灵活，性能好，功能强。

（4）“AppHost”（自宿主 WEB 应用程序托管服务）是 Jexus 独创功能，它的作用是将 asp.net core 或 node.js、tomcat 等“自宿主（自带 http 服务）”式的以控制台方式运行的 web 应用程序集成到 Jexus 工作序列，纳入 Jexus 统一管理，而无需进行反向代理，无需采用其它方式进行运行和守护；

（5）具备基于正则表达式的强大的“URL 重写”功能；

（6）支持 https，具有 SSL（TLS v1.0-1.3）加密数据安全传输能力；

（7）直接内置“入侵检测”安全模块，具有入侵检测和防御功能，能自动阻止已被识别的非法请求。这是 Jexus 有别于其它 Web 服务器的重要的特色功能之一；

（8）直接支持符合 OWIN 标准的 WEB 应用程序，如 Nancy、SignalR 等符合 OWIN 标准的应用框架，支持 WebSocket；

## 二、Jexus 的安装与更新

### （一）安装前的准备工作：

1、GLIBC 版本：Jexus 7.0/7.1 的编译环境是 CentOS7.9，对 GLIBC 版本的最低要求是 2.17，因此，Jexus 应该在 CentOS 7.9、Ubuntu 16.04 以上版本或者满足 GLIBC 版本要求的国产操作系统上运行。

2、依赖库：Jexus 使用了 .NET NativeAOT 技术，集成了 Mono 环境，因此，安装 Jexus 的 Linux 系统需要存在或安装 libicu、libkrb5 函数库。常用 Linux 一般默认已经安装了 libkrb6 或 libicu，部分系统需要手工安装。

3、管理员身份：Jexus 的安装、启动、停止等操作，操作者必须具备服务器管理员身份（即能使用 su 或 sudo 等命令提供 root 权限）。

### （二）在线安装：

指的是服务器能够连接互联网，服务器通过互联网从 Jexus 官方网站获取最新正式版安装包进行自动安装。安装命令是：

X86\_64（amd64）系统：

```
curl https://jexus.org/release/x64/install.sh|sh
```

AARCH64（arm64）系统：

```
curl https://jexus.org/release/arm64/install.sh|sh
```

*\* 强调：使用上述命令前应该确定自己当前身份是否是 root 身份。*

### （三）离线安装：

指的是从 jexus 官方网站下载 Jexus 安装包进行安装。这种安装方式常常用于服务器无法访问互联网的内网服务器。

Jexus 安装包是一个“tar.gz”压缩包，因此，“离线安装”过程，其实就是下载安装包，传输到目标服务器，解压到工作目录这样的操作过程。

1、下载：到 [www.jexus.org](http://www.jexus.org) 下载你需要的 Jexus 安装包。

2、将安装包上传到服务器的安装目录。Jexus 习惯上使用“/usr/jexus”作为工作目录，因此，建议你把安装包上传到“/usr”文件夹中。

3、解压：在 jexus 安装包所在文件夹中，以 root 身份解压 Jexus 压缩包（安装包）。如 x86\_64 是：`sudo tar -zxvf jexus-7.0.x-x64.tar.gz`

4、删除安装包。

5、从备份文件夹中恢复之前 Jexus 的配置文件（如果有）。

### （四）关于 Jexus 的更新或重装：

由于 Jexus 是绿色软件，所谓的“安装”其实就是下载和解压，因此，Jexus 的更新、重装，其实与上述安装过程是相似的，不同点就是在安装之前多两个步骤：

（1）停止 Jexus 运行；

（2）删除或更名（备份）之前的 Jexus 的工作文件夹。



（五）检查版本和初始化：

1、检查 Jexus 版本号的命令是 “`/usr/jexus/jws -V`”，如果 Jexus 安装正常，此命令将打印出 Jexus 的版本号以及集成的 .NET (mono) 运行时的版本号。

2、初始化命令是 “`sudo /usr/jexus/jws init`”，这是安装 Jexus 后的非必要操作，但是建议执行一次。

## 三、Jexus 的操作命令

### （一）基本操作命令：

Jexus 核心命令是“jws”，如果你把 jexus 安装在“习惯”位置，即“/usr/jexus/”文件夹中，那么，含绝对路径的完整命令将是“/usr/jexus/jws”。

“jws”其实是一个 shell 脚本文件，具体内容可以自行查看。

Jexus 的操作是通过为“jws”提供不同的操作参数实现的，包括“启动(start)”“停止(stop)”“重启(restart)”“运行状态(status)”“初始化(init)”“显示版本(-V)”。初始化和显示版本号已经在前文进行了说明，不再此处复述。

### 1、启动命令

#### （1）启动 Jexus 程序并同时启动所有的网站：

```
sudo /usr/jexus/jws start
```

#### （2）在 Jexus 运行期间单独启动一个网站：

```
sudo /usr/jexus/jws start 网站名
```

注：“网站名”在 Jexus 中，指的是网站配置文件名，jexus 的网站配置文件即是“网站名”。

## 2、停止命令

（1）停止 Jexus 程序同时停止所有的网站：

```
sudo /usr/jexus/stop
```

（2）停止一个指定的网站（web 应用程序）

```
sudo /usr/jexus/stop 网站名
```

## 3、重启命令

（1）重启 jexus 程序和所有的网站程序

```
sudo /usr/jexus/jws restart
```

（2）重启指定的网站程序

```
sudo /usr/jexus/jws restart 网站名
```

## 4、显示 Jexus 是否处理于运行状态

```
sudo /usr/jexus/jws status
```

注意：以上命令都加有 **sudo** 命令，指的是以管理员身份发起操作，如果你本来就是管理员身份，就不必使用 **sudo**。

（二）关于开机启动：

让 Jexus 实现开机启动，有两个办法（强调：只能选其中之一）：

方法一：通过“/etc/rc.local”进行。

具体方法是：在“/etc/rc.local”文件中加入“/usr/jexus/jws start”

一行。

此方法使用简便，但要注意：一是要把 `jws` 的路径和文件名写正确，要使用“`jws`”命令文件的绝对路径；二是 `rc.local` 文件要有“可执行”属性，建议运行一下这个命令：“`sudo chmod +x /etc/rc.local`”。

**方法二：**将 `jexus` 注册到 `systemd` 中成为系统服务。

具体方法如下：

- 1、注册服务：`sudo systemctl enable /usr/jexus/jws.service`
- 2、启动服务：`sudo systemctl start jws`
- 3、停止服务：`sudo systemctl stop jws`
- 4、重启服务：`sudo systemctl restart jws`
- 5、注销服务：`sudo systemctl disable jws`

**特别强调：**一旦 `jexus` 被注册为系统服务之后，`jexus` 的启动、停止、重启等操作只能使用 `systemctl` 进行操作，不能再使用不带网站参数的“`jws start`”“`jws stop`”“`jws restart`”等命令进行操作。

## 四、Jexus 的全局配置

在 jexus 的工作文件夹中（一般是“/usr/jexus”）有一个基本的配置文件，文件名是“jws.conf”，这是 Jexus 的全局配置文件。

主要配置项如下：

**SiteConfigDir:** 指定网站配置文件夹的路径，默认配置是 jexus 程序文件夹下的“siteconf”子目录。该项配置不建议修改。

**SiteLogDir:** 指定 jexus 日志和网站访问日志的存放路径。默认配置是 jexus 程序文件夹下的“log”子目录。该项配置不建议修改。

**Httpd.Processes:** 指的是 jexus httpd 工作进程的数量，默认为 2，设为 0 表示自动，即由 jexus 根据服务器 cpu 数量自行确定。

该项配置能影响 Jexus 处理 html 等静态文件以及数据转发等方面的性能，其具体数量应该结合服务器 cpu 数量、内存富余度和并发请求数量综合考虑，一般来情况下配置 2 个进程就行，如果服务器并发流量大，可以按 cpu 核心数的三分之一或二分之一设置。

**Httpd.MaxTotalMemory:** 指的是允许 Jexus 的 httpd 工作进程最多使用多少内存，单位为“兆字节”，设为 0 表示“自动”。

**Httpd.MaxCpuTime:** 表示 Jexus 的 httpd 工作进程最多能使用多少 cpu 时间，单位为秒，达到这个时间值后，jexus 的 httpd 工作进程就会自动重启。设为 0 表示禁用这个选项。

**Httpd.MaxConnPerIp:** 表示 jexus httpd 工作进程允许每个 IP 地址能同时发出多少个 TCP 连接，当某 IP 地址超过这个数量的连接时，Jexus 将拒绝接受。设为 0 表示禁用这个选项。

**Php-fcgi.Set:** PHP 运行环境设置，本设置的值分两部分，两个部分之间用英文逗号分开。第一部分是指定 php-cgi 命令文件的完整路径（一般是“/usr/bin/php-cgi”）；第二部分是设定 php-cgi 的最大工作进程数量。本项设置不是必须的，如果你服务器不通过 Jexus 直接运行 PHP WEB 程序，就可以禁用该项（在配置行前加“#”号）。

## 五、Jexus 的网站配置

Jexus Web Server 可以同时运行多个站点，没有数量限制。

一个具体的站点，由“端口”“域名”“虚拟根路径”三个基本要素确定，三个要素中任何一项不同，即表示不同的站点。

必须把所有网站配置文件放到全局配置文件“jws.conf”所指定的网站配置文件夹内（默认是 jexus 工作目录内的“siteconf”子文件夹）。这个文件夹除了网站配置文件，不能有其它任何文件，因为 jexus 会认为该文件夹中的任何一个文件都代表着一个具体的网站配置（子文件夹除外，不在 jexus 搜索之列）。

每个网站有且只有一个配置文件，配置文件的文件名代表这个网站的名称，因此配置文件的文件名应该尽量具有网站个性特点，以便管理员容易记忆和识别。要特别注意：文件名不能有空格！支持中文，但不建议使用。

一个网站可以拥有任意多的域名或“泛域名”，不同网站不能有相同的域名，每个端口最多只能有一个没有域名的网站（域名栏填“\*”号）。在 80 端口，这个没有域名的网站叫做“默认网站”。

（一）网站配置文件的基本配置项：

**port:** 站点服务端口，这是必填项。http 的标准端口是 80，https

的标准端口是 443。

**root:** 根路径。指定网站虚拟根路径（URL 根路径）和存放网站文件的物理文件夹（物理根）路径，两种路径用空格符分开。URL 根路径常常是 “/”。如果你的网站文件存放在 “/var/www/default” 这个文件夹中，那么，root 项就应该是 “root=/ /var/www/default”。

root 项是必填项。如果遇上在当前服务器并没有网站程序的“全站反向代理”这种情况，你可以在服务器单独创建一个空白文件夹并设定它，也可以设为 “/tmp”。

**hosts:** 配置网站域名。一个站可以指定多个域名，各域名间用英文逗号分隔。也可以配置“泛域名”，比如 “\*.mysite.com”。如果填 “\*” 号，就表示访问这个站可以是任意的域名或是服务器 IP 地址，即，区分这个网站不是用域名区分，而是用端口或虚拟根路径进行区分。

## （二）网站配置文件的拓展（高级）配置

一个网站配置了前述的三项基本选项一般就能正常运行了。本部分指的是基本配置之外的拓展（高级）配置，包括“反向代理”“URL 重写”“端口转发”“AppHost”“IP 地址黑（白）名单”等等。

具体说明如下：

### 1、默认首页（Indexes）：

Jexus 已经内置默认首页的定义，包括 index.html、index.htm、default.html、default.htm、index.aspx、default.aspx。如果你网站的首



页没有被包括在这些默认首页中，可以通过启用该选项指定你的默认首页。

## 2、URL 重写 (rewrite):

“URL 重写”是指 WEB 服务器将访问者向服务器发起的 URL 请求按指定的匹配规则解释和匹配到另外的一个真实 URL 路径资源上。比如，当访问者访问 “.php” “.asp” 等类型的文件时，服务器以 “/404.html” 这个 URL 应答，你可以设为：

```
rewrite=^/.+?\.(asp|php|cgi)$ /404.html
```

rewrite 格式：

“rewrite=”的后面是两部分组成，两部分之间由一个空格分开。

空格前是匹配的条件：用正则表达式描述 URL 的匹配条件。

空格后是匹配的目标：指的是如果用户访问的路径合乎前面的匹配条件时，服务器将以哪个规则回应。

如：

把 “/bbs” 匹配到 “/bbs/index.aspx”，把 “/bbs/file-编号” 匹配到 “/bbs/show.aspx?id=编号”：

```
rewrite=^/bbs$ /bbs/index.aspx
```

```
rewrite=^/bbs/file-([0-9]{1,6})$ /bbs/show.aspx?id=$1
```

格式解释：rewrite 的等号后含有两部分内容，用空格分开。前半部分是一个正则表达式，用于描述需要 URL 重写的（用户浏览器中的）url 路径样式，后半部分是当用户的 URL 合乎前面的正则表达式时，JWS 应该重写和访问的真实 URL 路径。

### 3、禁止或允许某 IP 或 IP 段访问网站（**denyfrom**、**allowfrom**）:

#### A、只允许某些 IP 地址访问网站（白名单功能）

默认情况下，允许所有 IP 地址访问。如果手工设置 IP 地址白名单，那么，白名单之外的 IP 地址会自动划入黑名单而禁止访问服务器。

配置格式，形如：

**allowfrom=1.2.3.\***           # 一个 IP 地址段

**AllowFrom=2.2.3.3**       # 一个具体的 IPv4 地址

#### B、禁止某 IP 或某 IP 段访问网站（黑名单功能）

默认情况下，本配置为空。如果手工添加需要禁止访问的 IP 地址（段），必须合乎一个规则：黑名单必须是白名单的真子集。

配置格式，形如：

**denyfrom=111.222.111.\***   # 一个 IP 地址段

**denyfrom=101.201.1.132**   # 一个 IP 地址

*注：为了不使配置复杂化，建议只使用“黑名单”。*

### 4、禁止访问某文件夹及其子文件夹中的内容（**DenyDirs**）：

格式是：**DenyDirs=PATH**

**PATH** 指的是相对于网站根文件夹的 URL 路径，如 “/abcfiles” 或 “~/abcfiles”。可以同时指定多个路径，用英文逗号分开。

### 5、是否对请求的 URL 参数等进行安全检测（**CheckQuery**）:

本选项的值包括 **true** 和 **false** 之一，默认是 **true**，即需要进行安全检查，除非你的确需要关掉这个选项，否则可以不填。格式如下：

CheckQuery=false

注：关掉本项可以提高服务器速度，但就安全而言，不建议关掉它。

## 6、NOFILE（无文件替代）功能：

本功能是 Jexus 的特有功能，指的是如果服务器找不到真实文件时，服务器将用使用一个什么文件去替代。格式是：

NoFile=PATH

PATH：指定的替代文件和相当于网站根文件夹的完整路径，如：

Nofile=/mvc/controller.aspx 或者 Nofile=/index.php

（提示：替换后，Jexus 会把原始请求的路径资源保存到一个名为“X-Real-Uri”的 HTTP 请求头中，以便目标文件作进一步处理）

## 7、关闭网站访问日志（NoLog）：

nolog=yes

（注：禁用网站日志功能会轻微提高 WEB 服务器系统的处理速度，但不足也是明显的，就是你无法详细了解网站的访问情况了）

## 8、长连接开关（Keep\_Alive）：

Keep\_Alive=true 或 false

注：其值为 true、false 或 yes、no，默认使用长连接（true），大多数情况可以不使用该选项，除非你认为真有必要禁止长连接。

## 9、反向代理功能（Reproxy）：

这是 Jexus 的一个重要功能。指的是当用户访问某个路径或任何路径时，Jexus 将这个请求转发给其它的服务器。

通过 Reproxy 功能，我们可以实现负载均衡或“高可用性”。

格式是：

Reproxy=请求路径 目标服务器 IP:端口，如：

Reproxy=/ 1.1.1.1:800 （全站反代）

Reproxy=/abc 1.1.1.1:900/abc （部分路径反代）

Reproxy=/ 1.1.1.1:5000, 2.2.2.2:5000 （多目标服务器反代）

值得强调的是，“多目标服务器反向代理”将产生负载均衡（WEB 集群）和高可用性效果，即，当用户访问该站时，Jexus 就会随机选择一台后端服务器进行处理，如果 Jexus 发现某台服务器已经宕机而没有应答，Jexus 就会把请求转发给其它剩余的服务器。

注意：当“部分反代”和“全站反代”多条规则同时出现时，一定要把“全站反代”那一条规则放到最后。

## 10、使用 FAST-CGI 提供的服务（FastCGI.Add）：

对于 TCP 连接的 fast-cgi：

fastcgi.add=php | tcp:127.0.0.1:9000

对于 unix sockets 连接的 FastCGI：

fastcgi.add=php,php3 | socket:/tmp/phpsvr

## 11、启用或关闭 gzip 压缩功能（UseGZIP）：

UseGzip=true 或 false

解释：Jexus 默认已经启用 gzip 压缩功能，所以，除非你希望禁止该功能，才需要显式地设置此选项。

## 12、指定网站的用户身份（User）：

格式是：User=用户名

“用户名”指的是 Linux 系统中已经存在的一个用户名称，比如 `www-data`。

默认情况下，jexus 将使用“root”作为网站工作进程的用户身份，这种身份的权限很高，为了网站更加安全，建议指定一个权限低的用户身份，而“`www-data`”是大多数 Linux 系统的内置的专为网站准备的一个用户名，其权限较低，适合指定给网站使用。

### 13、启用 HTTPS 进行 SSL 安全传输（UseHttps）：

要为网站启用 https，首先需设置 `UseHttps=true` 选项，表示该站启用 Https 安全连接。另外，对于 https，网站端口应该设为 443（https 默认端口）。

然后是设置 SSL 证书、私钥、协议版本、加密套件。

#### （1）设置证书文件：

`ssl.certificate`=完整的证书路径和文件名

#### （2）设置密钥文件：

`ssl.certificatekey`=密钥文件的完整路径和文件名

#### （3）协议版本：

`ssl.protocol`=版本列表

指的是 ssl 的版本号，包括：TLSv1.0、TLSv1.1、TLSv1.2 和 TLSv1.3。

可以同时多个版本号，各版本号之间用空格分隔。

该选项不是必填项，但为了安全起见，建议启用并选择 TLSv1.1、TLSv1.2、1.3 三个版本号。

(4) 加密套件:

```
ssl.ciphers=ECDHE-RSA-AES256-GCM-SHA384:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4:!DH:!DHE
```

本选项不是必填项。

#### 14、ASP.NET 网站专用配置:

(1) 配置 Asp.Net 工作进程数量 (默认为 1):

格式为: `AspNet.Workers=工作进程数`

默认情况下, jexus 为 Asp.Net 提供 1 个工作进程, 单进程的好处是节约内存, 但也有弱点, 比如, 难以充分发挥多 cpu 多核的性能优势, 大并发承受力、容灾力较多进程弱等等。

为 Asp.Net 开启“多进程”工作模式后, ASP.NET 网站的 web.config 中的 Session 状态服务不能再使用“inproc”模式, 而应该使用“StateServer”等其它模式, 否则容易出现 session 数据丢失问题。另外, 建议在 web.config 文件夹中配置“machineKey”项, 进一步提升 session 的可靠性。状态服务模式和主机键的设置方法与 IIS 相同。

设置 ASP.NET 多进程数量并非越多越好, 要考虑服务器 CPU 核数、可用内存、并发压力等多方面因素。一般来说, 进程数量配置为 CPU 核数的一半就行了, 最多不要超过 CPU 的核心数。

(2) 限制 ASP.NET 消耗 CPU 的时间数 (默认为 0, 无限制):

格式是: `AspNet.MaxCpuTime=单位为秒的时间值`

配置该选项后, 当该网站使用 cpu 的时间累积到你指定的最大值之后, Jexus 将重启这个网站。

(3) 限制 Asp.Net 使用内存的最大值（默认为 0，无限制）：

格式是：AspNet.MaxMemory=以兆为单位的内存数

设置该选项后，当你的网站程序使用的内存超过指定的最大值时，Jexus 将重启该站，释放之前占用的内存空间。

### 15、WEB 应用程序端口转发 (AppHost.Port)：

这是 Jexus 特有功能。

格式是：AppHost.Port=端口号

本项配置指的是将当前网站的请求数据全部转发到本机某 WEB 应用程序侦听端口，比如把请求转发到 Asp.net Core 应用程序的 5000 端口上。

可以使用多个端口，各端口号之间用英文逗号分开。

端口转发与反向代理功能相近，但端口转发的性能更高。

注意：一是端口转发不能用于虚拟根路径不是“/”的网站；二是端口转发只能在同一服务器上进行。

### 16、自宿主 WEB 应用程序托管服务 (AppHost)：

“自宿主 web 应用程序托管”是 Jexus 的一项独有的重要功能，它为 Asp.Net Core、Node.js、Tomcat 等自宿主 Web 程序提供了简单可靠、可控的一体化集成管理。

格式：AppHost={Cmd=命令行; AppRoot=工作路径; AppPort=端口号}

也可以分行书写(分行书写时，AppHost 和 “={” 必须在同一行)，例如：

```
AppHost={  
    Cmd=命令行;  
    AppRoot=应该程序文件夹路径;  
    AppPort=应用程序端口号;  
}
```

子配置项说明:

**Cmd:** 必选项。表示启动这个 web 应用程序的命令和命令参数, 如 `Cmd=dotnet /var/www/mysite/webapp.dll`。

**AppRoot:** 非必选项。表示这个应用程序的工作目录, 如: `AppRoot=/var/www/mysite`。如果本项没有指定, Jexus 将使用网站 root 项中的根文件夹路径作为应用程序路径。

**AppPort:** 必填项。表示这个应用程序使用的侦听端口, 多个端口用英文逗号分隔。

**Env:** 可选参数。表示这个应用程序工作时需要的环境变量, 如 `Env=(PATH=/myhost/bin:$PATH)`, 多个设置用英文逗号分开。

**ErrLog:** 可选项。表示将这个应用程序的异常输出重定向到指定的文件 (需填写完整路径);

**OutLog:** 可选项。表示将这个应用程序的控制台输出重定向到指定的文件 (需填写完整路径);

**User:** 可选项。以指定的用户身份运行该应用程序, 默认为 root 身份。

注意:



- 1、AppHost 像 AppHost.Port 一样，不支持虚拟路径；
- 2、AppHost 功能是将指定的具有 http 服务能力的 web 应用程序纳入 jexus 工作进程进行管理，对 Asp.Net Core 或 Node.js 等自宿主 web 程序用于生产环境具有重要的意义。

## 六、关于

### （一）重要提示：

1, Jexus v5.x/6.x 属免费软件，所有用户可以自由下载、传播和使用。但 Jexus 作者不对 Jexus 的用途、作用、效果、技术支持以及其它相关内容作任何明确或暗含的承诺，不承担任何直接或间接的责任。

2, Jexus v7.0 及更高版本，允许个人或者 10 人以下的小型团队免费使用，也允许开发者作为测试环境免费使用。其它情况下使用，原则上需要获得 Jexus 作者刘冰先生（宇内流云）的商业授权。

3, Jexus 作者对所有 Jexus 版本提供有偿的技术支持或定制开发。

4, 为了让 Jexus 可持续发展，欢迎广大用户及热心朋友对作者提供力所能及的支持，包括但不限于：

（1）您可以利用各种机会宣传 Jexus 优良品质、实用价值及成功案例，分享使用心得和技术理论，为推广、普及 Jexus 付出努力；

（2）您可以为初学者解难答疑；

（3）你可以将 Jexus 的一些重要的技术文档翻译为英文或其它文字并公开、免费发布，为 Jexus 跨出国门走向国际添砖加瓦；

（4）您可以利用开发新工程或升级旧工程的机会，利用 Jexus 把原来部署在 windows 上的 web 应用部署到 Linux 上，拓展 Jexus 的

使用量；

（5）您可以通过 Jexus 社区、QQ 群、邮箱等各种渠道及时反馈 Jexus 的 BUG，或者向 Jexus 的开发者、维护者提供富有建设性意义的各种建议；

（6）您可以为 Jexus 提供广告支持，或向开发者提供适量的经费赞助，为 Jexus 的持续发展提供动力源泉。

（二）官网、社区及联系方式：

官方网站：[www.jexus.org](http://www.jexus.org)

技术社区：[www.linuxdot.net](http://www.linuxdot.net)

QQ 群号码：700864320、102732979

邮箱地址：[j66x@163.com](mailto:j66x@163.com)