

# 安装OpenResty

## 1. 安装

首先你的Linux虚拟机必须联网

### 1) 安装开发库

首先要安装OpenResty的依赖开发库，执行命令：

```
yum install -y pcre-devel openssl-devel gcc --skip-broken
```

### 2) 安装OpenResty仓库

你可以在你的 CentOS 系统中添加 `openresty` 仓库，这样就可以便于未来安装或更新我们的软件包（通过 `yum check-update` 命令）。运行下面的命令就可以添加我们的仓库：

```
yum-config-manager --add-repo  
https://openresty.org/package/centos/openresty.repo
```

如果提示说命令不存在，则运行：

```
yum install -y yum-utils
```

然后再重复上面的命令

### 3) 安装OpenResty

然后就可以像下面这样安装软件包，比如 `openresty`：

```
yum install -y openresty
```

### 4) 安装opm工具

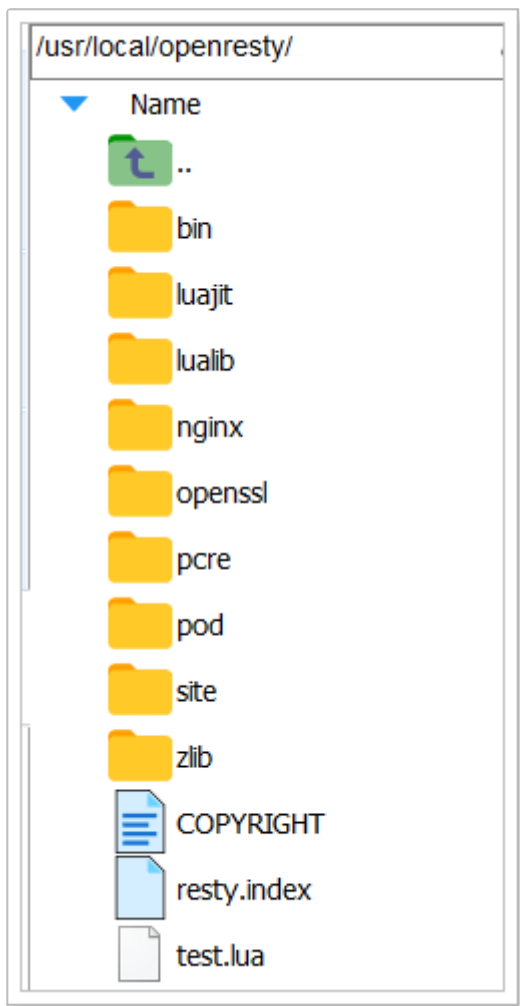
opm是OpenResty的一个管理工具，可以帮助我们安装一个第三方的Lua模块。

如果你想安装命令行工具 `opm`，那么可以像下面这样安装 `openresty-opm` 包：

```
yum install -y openresty-opm
```

### 5) 目录结构

默认情况下，OpenResty安装的目录是：`/usr/local/openresty`



看到里面的nginx目录了吗，OpenResty就是在Nginx基础上集成了一些Lua模块。

## 6) 配置nginx的环境变量

打开配置文件：

```
vi /etc/profile
```

在最下面加入两行：

```
export NGINX_HOME=/usr/local/openresty/nginx  
export PATH=${NGINX_HOME}/sbin:$PATH
```

NGINX\_HOME：后面是OpenResty安装目录下的nginx的目录

然后让配置生效：

```
source /etc/profile
```

## 2. 启动和运行

OpenResty底层是基于Nginx的，查看OpenResty目录的nginx目录，结构与windows中安装的nginx基本一致：

```
[root@node1 nginx]# pwd
/usr/local/openresty/nginx
[root@node1 nginx]# ll
总用量 4
drwxr-xr-x. 2 root root 4096 8月 10 21:45 conf
drwxr-xr-x. 2 root root 40 8月 10 21:29 html
drwxr-xr-x. 2 root root 41 8月 11 10:05 logs
drwxr-xr-x. 2 root root 19 8月 10 21:29 sbin
```

所以运行方式与nginx基本一致：

```
# 启动nginx
nginx
# 重新加载配置
nginx -s reload
# 停止
nginx -s stop
```

nginx的默认配置文件注释太多，影响后续我们的编辑，这里将nginx.conf中的注释部分删除，保留有效部分。

修改 `/usr/local/openresty/nginx/conf/nginx.conf` 文件，内容如下：

```
#user nobody;
worker_processes 1;
```

```
error_log logs/error.log;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    server {
        listen 8081;
        server_name localhost;
        location / {
            root html;
            index index.html index.htm;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}
```

在Linux的控制台输入命令以启动nginx:

```
nginx
```

然后访问页面: <http://192.168.150.101:8081>, 注意ip地址替换为你自己的虚拟机IP:

## 3. 备注

加载OpenResty的lua模块:

```
#lua 模块
lua_package_path
"/usr/local/openresty/lualib/?.lua;;";
#c模块
lua_package_cpath
"/usr/local/openresty/lualib/?.so;;";
```

common.lua

```
-- 封装函数, 发送http请求, 并解析响应
local function read_http(path, params)
    local resp = ngx.location.capture(path, {
        method = ngx.HTTP_GET,
        args = params,
    })
    if not resp then
        -- 记录错误信息, 返回404
        ngx.log(ngx.ERR, "http not found, path: ",
path , ", args: ", args)
        ngx.exit(404)
    end
    return resp.body
end
-- 将方法导出
local _M = {
    read_http = read_http
}
return _M
```

释放Redis连接API:

```

-- 关闭redis连接的工具方法，其实是放入连接池
local function close_redis(red)
    local pool_max_idle_time = 10000 -- 连接的空闲时间，单位是毫秒
    local pool_size = 100 --连接池大小
    local ok, err =
red:set_keepalive(pool_max_idle_time, pool_size)
    if not ok then
        ngx.log(ngx.ERR, "放入redis连接池失败：",
err)
    end
end

```

读取Redis数据的API:

```

-- 查询redis的方法 ip和port是redis地址，key是查询的key
local function read_redis(ip, port, key)
    -- 获取一个连接
    local ok, err = red:connect(ip, port)
    if not ok then
        ngx.log(ngx.ERR, "连接redis失败：", err)
        return nil
    end
    -- 查询redis
    local resp, err = red:get(key)
    -- 查询失败处理
    if not resp then
        ngx.log(ngx.ERR, "查询Redis失败：", err, ",
key = " , key)
    end
    --得到的数据为空处理
    if resp == ngx.null then
        resp = nil
        ngx.log(ngx.ERR, "查询Redis数据为空，key = ",
key)
    end
    close_redis(red)
    return resp
end

```

开启共享词典：

```
# 共享字典，也就是本地缓存，名称叫做：item_cache，大小150m  
lua_shared_dict item_cache 150m;
```