快速上手

1. 如何通过 njet 部署 WEB SERVER

以下是一个使用 Njet 部署 Web Server 的完整示例：

1.1 安装 Njet

首先，请参照 njet 使用手册的说明安装 njet。

1.2 配置 Njet

Njet 的主要配置文件为 njet.conf。可以通过修改该文件来配置 Njet。

例如，以下是一个简单的 Njet 配置文件示例，用于将所有请求重定向到一个 HTML 文件：

```
Go
http {
    server {
        listen 80;
        server_name example.com;

        location / {
            root /var/www/html;
            index index.html;
        }
    }
}
```

上述配置中，我们在 HTTP 块中定义了一个名为“server”的服务器块。该服务器块监听 80 端口，并将请求的根目录设置为/var/www/html。如果请求的路径不存在，默认会返回 index.html 文件。

1.3 部署 Web 应用程序

在配置 Nginx 之前，需要将 Web 应用程序部署到服务器上。可以将 Web 应用程序放

置在服务器上的任何位置，只要在 Nginx 配置文件中正确设置 root 目录即可。

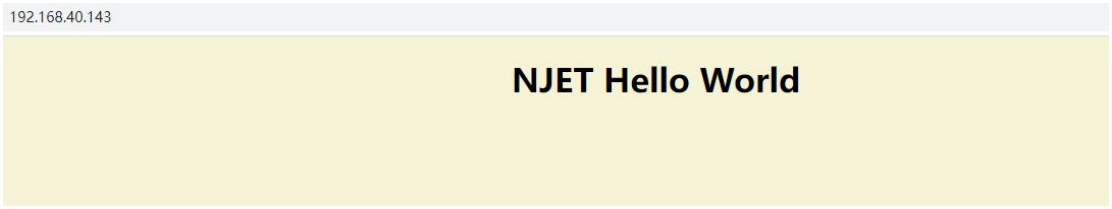
1.4 启动 njet

在完成 Njet 配置后，可以通过以下命令启动 Njet:

```
Bash
  njet  -p /tmpr/njet/      -c conf/njet.conf
  常见启动参数：
      -p 指定 prefix 配置文件路径，不指定，默认/etc/njet
      -c 指定配置文件，不指定，默认 njet.conf
      -e 指定 error 日志文件
```

1.5 访问 Web 应用程序

现在，可以使用 Web 浏览器访问 Web 应用程序。只需输入服务器的 IP 地址或域名即可访问 Web 应用程序。如果您按照上述示例配置 Njet，则应将 Web 应用程序放置在 /var/www/html 目录中，并使用服务器的 IP 地址或域名访问它。



总之，上述步骤为您提供了一个基本的示例，您可以根据需要进行修改和定制。在实际部署 Web 应用程序时，可能需要更复杂的 Njet 配置，例如反向代理、负载平衡等。

2. 如何通过 njet 部署反向代理 server

njet 支持反向代理功能的使用，以下是反向代理功能实现的一个完整实例:

2.1 安装 njet

首先，请参照 njet 相关说明章节安装 njet。

2.2 配置 njet.conf 配置文件

如下代码块所示，我们想要使用反向代理功能，需要使用 proxy_pass 指令来进行反向代理服务，proxy_pass 设置连接被代理服务器的协议、IP 地址或套接字，也可以是

域名或 upstream 定义的服务器组，这里我们使用 upstream 作为上游资源服务器组。

```
JSON
http {
    upstream backend1{
          server 127.0.0.1:8080;
          server 127.0.0.1:8081;
        }
    server {
        listen 9000;
        location /{
               proxy_pass http://backend1;
        }
    }
}
```

2.3 启动 njet，指定配置好的配置文件

在完成 Njet 配置后，可以通过以下命令启动 Njet:

```
Bash
  njet  -p /tmpr/njet/      -c conf/njet.conf
  常见启动参数：
      -p 指定 prefix 配置文件路径，不指定，默认/etc/njet
      -c 指定配置文件，不指定，默认 njet.conf
      -e 指定 error 日志文件
```

2.4 请求上游资源，验证反向代理服务

如果按照上述步骤进行部署与配置后，我们可以在 Centos 服务器命令行界面使用 curl 命令来请求上游服务器的资源，以此来验证反向代理服务是否部署成功。

```
Bash
curl http://127.0.0.1:9000/test/hello.html
<!doctype html>
<html lang="en">
 <head>
  <title>Document</title>
 </head>
 <body>
    Njet Hello World!
```

```
 </body>
</html>
```

如上方请求结果所示，请求上游服务器资源成功，成功通过反向代理服务请求到所需资源，可以根据具体需求来进行配置和使用。

3. 如何使用 njet 的动态配置功能

使用 njet 动态配置功能有以下几种方式。

3.1 直接通过 curl 的方式

以动态黑白名单为示例，执行命令:

```Plaintext
curl -X GET http://127.0.0.1:8081/config/1/config/http_dyn_bwlist
```

可以得到当前包括动态和静态的黑白名单配置:

```JSON
{
  "servers": [
    {
      "listens": [
        "0.0.0.0:90"
      ],
      "serverNames": [
        "localhost"
      ],
      "locations": [
        {
          "location": "/"
        },
        {
          "location": "/test_bwlist"
        }
      ]
    }
  ]
}
```

此时 njet 中没有添加任何的黑白名单。

如果需要在某一路径下添加一个黑白名单,执行命令:

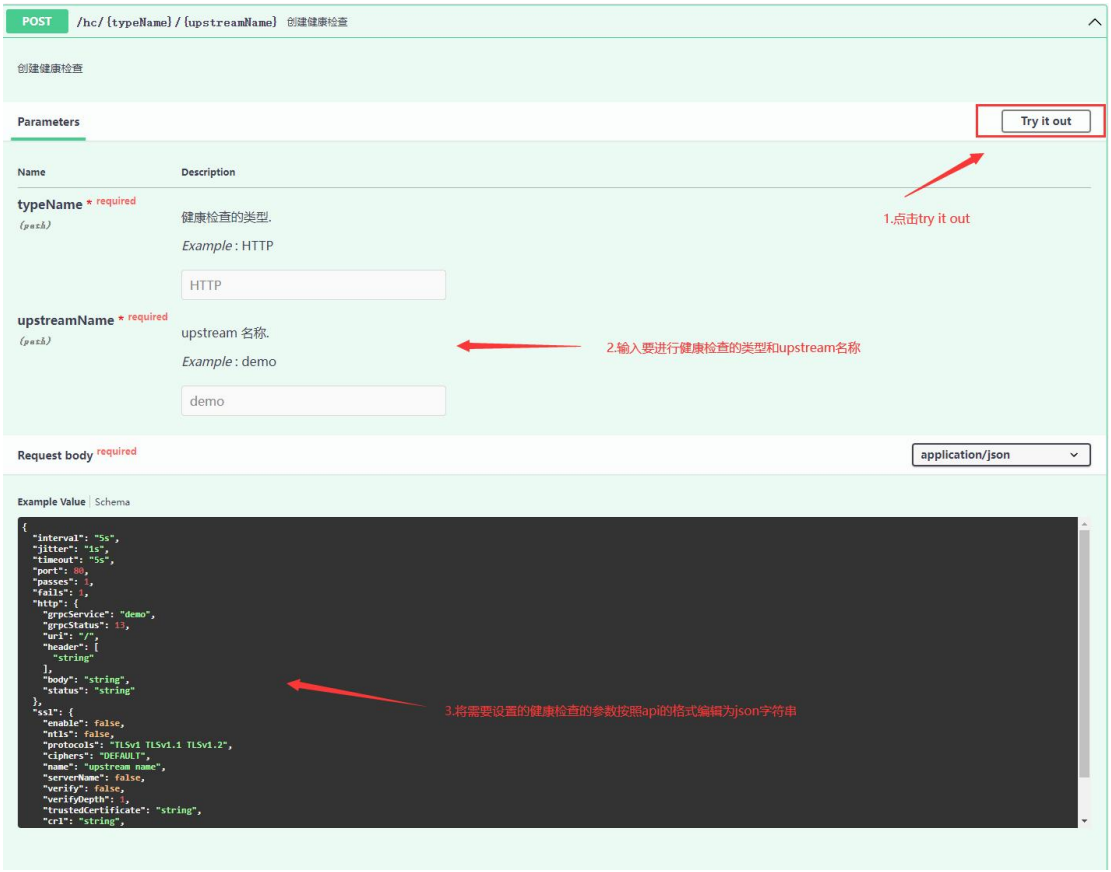
```HTTP
curl -X PUT
http://192.168.40.119:8081/config/1/config/http_dyn_bwlist \
-d '{
       "servers": [
         {
           "listens": [
             "0.0.0.0:90"
           ],
           "serverNames": [
             "localhost"
           ],
           "locations": [
             {
               "location": "/"
             },
             {
               "location": "/test_bwlist",
               "accessIpv4":
                      {
                      "rule": "deny",
                      "addr": "192.168.40.118",
                      "mask": "255.255.255.255"
                }

             }
           ]
         }
       ]
}'
```

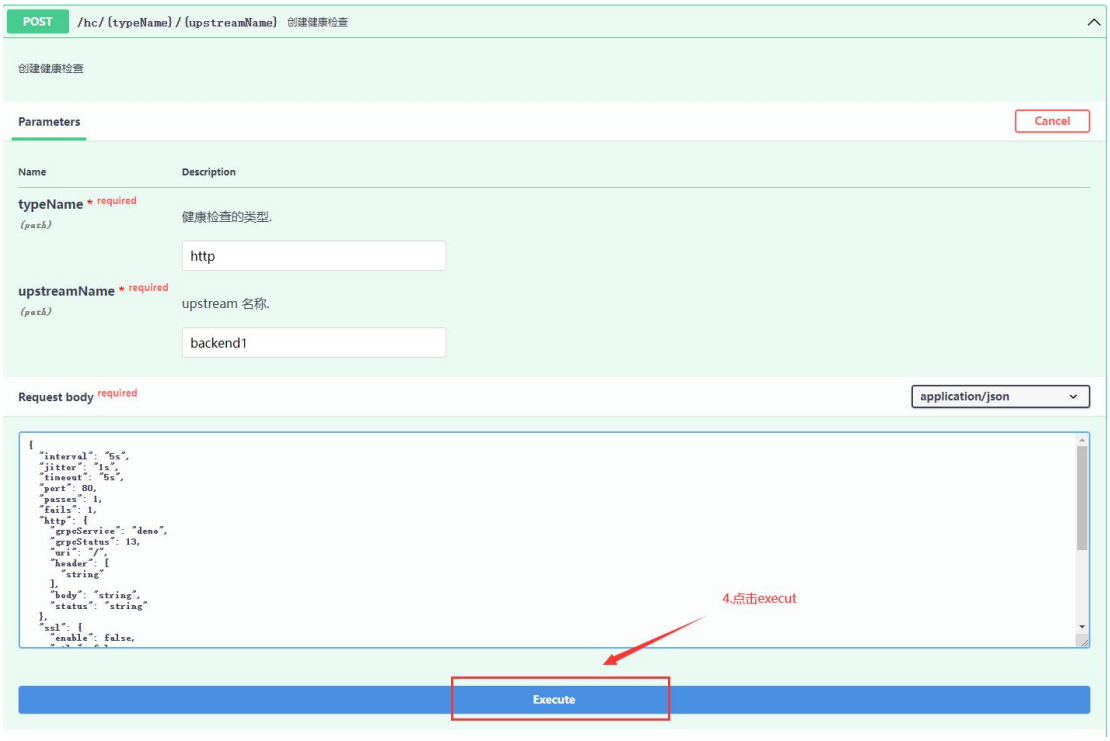
3.2 通过 Swagger 的进行动态配置

通过 swagger 的 url，http://njetaddr:8081/doc/swagger/ 进入 swagger 页面

以 helper 进程主动健康检查为例，按照下图示例中内容，编辑好需要配置的 json 内容。



点击 execute，使用编辑好的 json 数据调用该 api。



3.3 通过 GUI 的进行动态配置

通过 GUI 页面，url:http://njetServIP:8081/doc/gui/ 进入 gui 页面。

按照下图示例中，以 http_split_clients_2 为例，修改参数后点击保存，配置即可生效。