

How to setup a webserver for TR-143 Performance Tests

[TR-143](#) defines the standard framework for running performance tests from a CPE (Customer Premises Equipment) through the TR-069 protocol. Specifically for the Download and Upload tests, the CPE requires a functional HTTP server to either download or upload files to compute current throughput and measure network performance.

This guide explains how to configure a web server that supports both download and upload functionality for TR-143 performance testing when using CONTROL.

Requirements

- Linux server (this guide uses Ubuntu 24.04 LTS)
- Root or sudo access
- Network connectivity between the CPE and the server

Installation Steps

Install Nginx

Update the package list and install Nginx:

```
sudo apt update
sudo apt install nginx
```

Create Directory Structure

Create the necessary directories for downloads, uploads, and temporary files:

```
sudo mkdir -p /var/www/files
sudo mkdir -p /var/www/files/download
sudo mkdir -p /var/www/files/upload
sudo mkdir -p /var/www/files/tmp
```

Set the correct ownership for the web server:

```
sudo chown -R www-data:www-data /var/www/files
```

Configure Nginx

Edit the Nginx configuration file located at `/etc/nginx/sites-available/default` with the following content:

```
server {
    listen 80;
    server_name noname;
    root /var/www/files/download;

    location / {
        autoindex off;
    }

    client_max_body_size 100M;

    location /upload {
        client_body_temp_path /tmp;
        alias /var/www/files/upload;
        dav_methods PUT;
        dav_access user:rw group:rw all:r;
    }
}
```

Configuration notes:

- `listen 80` — Server listens on HTTP port 80
- `root /var/www/files/download` — Root directory for download files
- `client_max_body_size 100M` — Maximum upload file size set to 100MB
- `dav_methods PUT` — Enables HTTP PUT method for file uploads
- `location /upload` — Dedicated endpoint for upload operations

Apply Configuration

Verify the configuration syntax:

```
sudo nginx -t
```

If the configuration is valid, restart Nginx to apply changes:

```
sudo systemctl restart nginx
```

Testing the Setup

Download Test

To test download functionality, first create a test file in the download directory (e.g., `100MB.bin`), then download it from a client machine:

```
wget http://<server_ip_address>/100MB.bin
```

Upload Test

To test upload functionality, upload a file from your local machine to the server (uploading a file named `**`test-upload.bin`**` from your local host to the server):

```
curl -X PUT --data-binary @test-upload.bin http://<server_ip_address>/upload/test-upload.bin
```

Replace `` with your server's actual IP address and ensure the test file exists locally before running the upload command.

Troubleshooting

- Verify Nginx is running: ``sudo systemctl status nginx``
- Check Nginx error logs: ``sudo tail -f /var/log/nginx/error.log``
- Ensure firewall allows HTTP traffic on port 80
- Verify file permissions in ``/var/www/files`` directories

Revision #2

Created 2026-02-13 22:40:47 UTC by ipena@zequenze.com

Updated 2026-04-09 03:15:52 UTC by mauro@zequenze.com