

# Count number of HTTP threads in CLI

You may want to check the traffics in your server prom time to time, and checking CPU and number of threads are actually essential.

You can simply check the number of HTTP threads on terminal:

```
ss | grep http | wc -l
```

Below code (httpd\_count.sh) prints the total number of HTTP threads, so you can check its status constantly.

## httpd\_count.sh

```
#!/bin/bash
for i in {1..99999}
do
    ss | grep http | wc -l
    sleep 1
done
```

Below code (httpd\_count.php) prints the total number of HTTP threads with # bar, so you can check its status more visually.

## httpd\_count.php

```
#!/usr/bin/php
<?php
// HTTP thread monitoring - Chun Kang (ck@ckii.com) at 2021-11-02

foreach($argv as $v)
{
    $v = strtolower($v);
}

$loop_limit=99999;

$hostname = gethostname();

for($i=0; $i<$loop_limit; $i++)
{
    if ($i) sleep(1);

    $resp = shell_exec( "ss | grep http | wc -l");
    $process_count = intval($resp);

    $cpu_usage = trim( shell_exec( "grep 'cpu ' /proc/stat | awk '{usage=($2+$4)*100/($2+$4+$5)} END {print usage \"%\"}'" ) );

    if ($process_count<100) $bar_str = "\033[32m";
    else if ($process_count<200) $bar_str = "\033[33m";
    else if ($process_count<500) $bar_str = "\033[35m";
    else
    {
        $bar_str = "\033[31m";
    }
    $bar_str = "";
    if ($process_count)
    {
        $bar_count = round( $process_count/30, 0);
        while( $bar_count>0 )
        {
            $bar_str .= "#";
            $bar_count--;
        }
        if ($process_count) $bar_str .= "#";
        $bar_str .= "\033[0m ";
    }

    echo "[{$hostname}] {$bar_str}" . number_format($process_count) . " (CPU {$cpu_usage})\n";
}
```