Stats and Logging

SLQ Wiki Fabrication Lab 2024/10/05 23:12

Stats and Logging

By default dokuwiki tracks changes and stores metadata inside /data/meta and /date/media-meta. This is accessible through various dokuwiki plugins. For example the changes plugin....

Recent Changes

- ARCHIVED: 3D printing inductions, Character & Diorama
- Handtools Induction Practical Exercise
- Edge Lit Acrylic
- Cheapduino Wearable Nightlight
- Electronics 101
- Tabletop Christmas Tree with LEDs
- Vibrobots
- 2022 Arduino Christmas Tree
- Pixel Box V2
- CAD 101 in Fusion 360 [Mick's version of the workshop]

Using Access Logs

If we want more detail, we can use the logstats plugin to generate a server log for our dokukwiki.

Then we can use goaccess to generate some pretty pictures, as well as export boring CSVs/JSON. As a command line tool, goaccess will also take input from pipes, which lets us use POSIX utilities to get what we want.

Installing Goaccess

Follow the instructions on the go access site to install from a repo

```
$ echo "deb http://deb.goaccess.io/ $(lsb_release -cs) main" | sudo tee -a
/etc/apt/sources.list.d/goaccess.list
$ wget -0 - https://deb.goaccess.io/gnugpg.key | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install goaccess
```

Config File and Browser List

The config file default installs to /etc/goaccess/goaccess.conf but for some reason, goaccess expects it in /etc/goaccess.conf. Copy it over and to the same for the browser.list file while you are at it. Set



the time and date formats to Apache/NGINX and the log type to COMBINED and we should be good to start.

Parsing the access.log

In this case I've copied the access log from /data/meta/access.log to work on it. We want:

- quarterly log
- bots and crawlers removed
- no internal IP address

so we are going to use a combination of tools for this. First up, lets use sed to grab the date range we want

sed -n '/1\/Jul\/2019/,/30\/Sep\/2019/ p' access.log

3/5

then grep with the -v option to exlude bots and dynomapper (this could be a single grep)

grep -i -v --line-buffered 'bot' | grep -i -v --line-buffered 'dyno'

Finally let run goaccess, excluding our local IP range, ignoring crawlers and output to a html file.

goaccess -e 192.168.0.0-192.168.254.254 --ignore-crawlers -a -o
q1report.html

Running all these commands piped (for the next quarter) we get:

```
sed -n '/1\/0ct\/2019/,/31\/Dec\/2019/ p' access.log | grep -i -v --line-
buffered 'bot'| grep -i -v --line-buffered 'dyno' | goaccess -e
192.168.0.0-192.168.254.254 --ignore-crawlers -a -o q2report.html
```

To see these logs just copy the resulting html to your web server root. You will end up with something like:



AB Dashboard											Last Updated: 2010-12-09 17:02:34 +1000	
lat Total Respects 148,886		at Veto Reports 93,977	ial Falect Requests O	latint, Proc. Time 5 secs		36	lat Unique Visitore 36,449			lat. Requested Files 2,406		
se Det 1910s 50,046		at Adverses O	at hier Found 1,214	M State Files 200		0	let tag ten 0 Byte			ist To Amount 123.23 GiB		
LRACUE VISITORS PER DAY INT MARE TO EXAMPLE VIEW INT MARE TO EXAMPLE VIEW INT MARE TO EXAMPLE VIEW					REGUESTED FILES (UFLS) Previous the control of the previous o							
• •	ta # Vialdora # Tx.	Amount 9 Data *			Hits *	Visitors P	TL Amoun	• Method	Protocol 9	Data P		
	III) 412(1.174) 1.46	GAB (1.125) ZHONGOOM			0/01/07/1000	0/01/01/2008	-					
	100 400(1270) 20.0	MB (0.27%) 285kg/0978					27-40 MB (5-11		HTERLE	stat		
		OB (1.12%) 275692078					10.0 MB (0.0			tothersq.vit	PERMI	
4 1,104(7		COR (1111) 20060000			3,239 (1.1710)	3,148 (1.070)	370 MB (513	13 DET	HTERLE	April (Marcold	echnology_tecourcechoorteacad	•
	100 ST0(1576) 18	COR (2005) ZOCHOVER					BAMB (S.O.			tomore we	PERMI	
1 (386)	ane 9/2(15m) 182	COR (1.11%) 24CHp2018			1380 (1.4710	148 (1.1110)	2.55 MB (0.02	5) DET	HTERLI	facilities/bankad	A STREET, STREE	
	Conceptual Anno Conceptual Anno Conceptual Anno Conceptual Anno Conceptual Anno Conceptual Concente Conceptual Conceptual Concente	Describer de la construir	Describered Version Version	Describoard OverAvL ANALYZED RECULERTS If Statistics If Statistic	Deshboard OverAuL ANNUYZED RECUESTS If Unit Road 148,886 93,977 16,00 12,00	Bashboard CoreALL ANNLYZED RECUESTS If Cale Repairs 148,886 193,977 10	Bits Name Second Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic Mit Single Republic	Bitschiboard ONEPALL ANALYZED RECULESTS Mitschilder 93,977 Mitschilder 93,977 Mitschilder 93,977 Mitschilder 0 Mitschilder 93,977 Mitschilder 0 <	Bit Deshiboard OVERALL ANALYZED RECUESTS Mit Strekensen 148,898 93,977 16 bet drive 0 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12 bet drive 12	Bit Deshiboard OVERALL ANALYZED REQUESTS Mit Deskingen jaf wirdingen jaf wirdingen	Bitshboard OFENLI ANALYZED RECUESTS W Statistics W Statistics 148,898 93,977 0 If alad Resett 148,898 93,977 0 If alad Resett 1211 1211 0 If alad Resett 0 If alad Resett If alad Resett 0 If alad Resett If alad Resett If alad Resett 0 If alad Resett	

This is pretty, but a more useful output would be csv. A csv output can be expand to include any number of record (in the config) so we can use it to get a sense of the static files downloaded, which can also be set to include only the types of files we are interested in (also in the config). To get csv output, just change the filetype of the output i.e q2report.csv

Accessing Metadata

To work out how many pages have been created, we need to go back to dokuwiki's metadata. What we want is the metadata stored in .changes file for each page and media file, that was created in our date range, and <u>not</u> created by one of our team. The changes file on an newly created page inside /data/meta/ looks like:

1487560052 192.168.6.90	С	workshops	user	created	18942
1.0,000002 102.100.00	•	nornopo		0.00.00	20012

"1487560052" is the timestamp in unix time the second is the IP address, "C" means created, and "user" is our users name. Thats all we need.

We only need the first line of each file called .changes. We can do this with the head command.

head -1 ./*changes

Next we want to narrow our selection to files created in our date range. A quick check of https://www.epochconverter.com/ will give us the date range we want, which is 1569852000 - 157780079. We can use awk to match this.

awk '(\$1+0)>1569852000 && (\$1+0)<1577800799'

Then we want to filter out our internal users with grep, and use the -c option to tally the output.

grep -v -c --line-buffered user



2024/10/05 23:12

Finally lets turn on the globstar in our shell so we can use head recursively.

shopt -s globstar

Now our piped commands look like this:

```
head -1 **/*.changes | awk '($1+0)>1569852000 && ($1+0)<1577800799'| grep - v -c --line-buffered mick
```

This gives use the pages created in the date range specified. Do find the media created, we can run the same command in the /data/media-meta directory, grepping for our media types.



Stats and Logging