

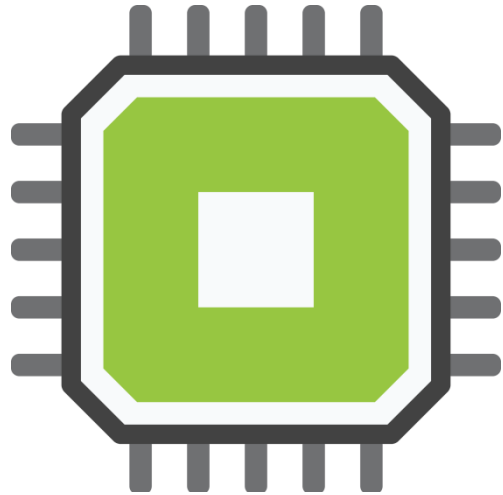
Analyze Web Logs with awk



Andrew Mallett

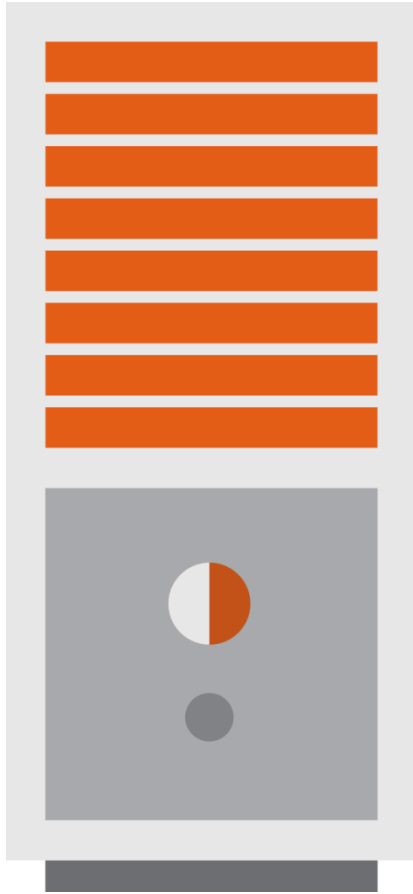
@theurbanpenguin | www.theurbanpenguin.com

DIY 'r' Us Analyze Web Logs



- Real life examples using associative arrays
- Process production log file of 30K lines
- Produce reports with awk

Processing Large Files

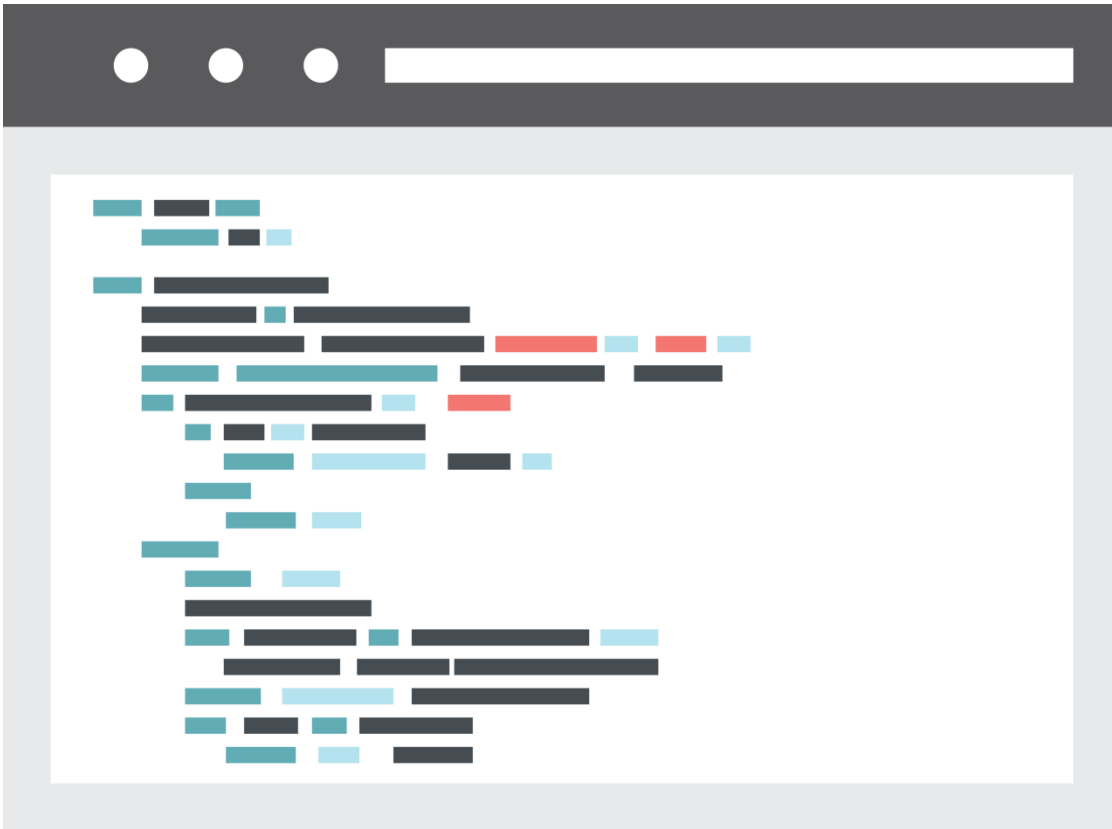


Tools such as sed and awk were written to handle large text files
Working with trivial files does not do justice to the commands

Danny has shared access logs of the DIY 'r' Us web servers with over 30K lines

Danny wants to learn how to create effective reports from these logs

Web Access Logs



Field 1 : Client IP

Field 2/3 : - - (identd and userid)

Field 4/5 : Time and time zone

Field 6/7/8 : Method, file, protocol

Field 9 : Status code

Field 10 : Size

```
$ awk ' { print $1 } ' access.log  
$ awk ' { print $9 } ' access.log
```

Understanding logs

Print the Client IP from the log

Print the status code from the log



Demo Time: Investigating the Web Logs

```
$ cat count.awk
BEGIN { FS=" "; print "Log access" }
{ ip[$1]++ }
END { for (i in ip)
print i, " has accessed ", ip[i], " times."
}
```

Array

Key

Value

Example: ip[192.168.0.1]3

Count unique accesses by a client

An array named **ip** is created that stores a **key** for each IP address

The **value** of the key is **incremented** each time the IP address is found

Using \$9 we could display the
count of all status codes such as
404 from the access logs



Demo Time: Analyzing Web Logs

Maximum Browser Count

```
BEGIN { FS=" "; print "Most Popular Browser" }  
{ browser[$12]++ }  
END { for ( b in browser )  
if ( max < browser[b] ) {  
    max = browser[b];  
    maxbrowser = b; }  
print "Most access was from ", maxbrowser, " and ", max, "  
times." }
```

Summary



- Used awk arrays
- Used loops within awk
- Used if statements within awk
- Analyze web logs
- You now know grep, sed, and awk

Put what you have learned
from Danny to use in your own
organization