



Thomas Watson

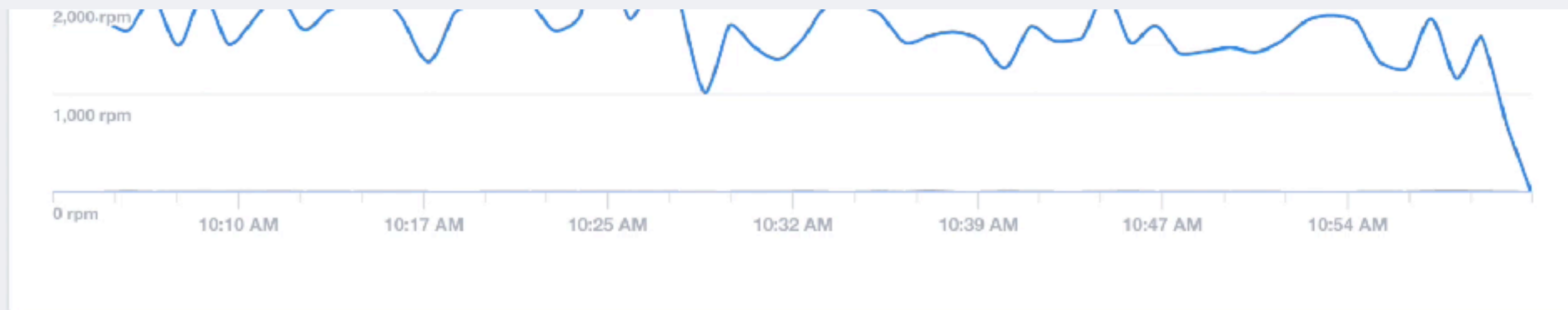
@wa7son

[github.com/watson](https://github.com/watson)





opbeat



Endpoints		Avg. resp. time	95th percentile	RPM	Impact ▲
GET /coffee		107 ms	238 ms	670.13 rpm	<div></div>
GET /roast/{id}([a-f0-9]{24})		118 ms	148 ms	534.37 rpm	<div></div>
<u>GET /coffee-level</u>		652 ms	215 ms	16.87 rpm	<div></div>
GET /roast/favorites/count		1,021 ms	125 ms	9.1 rpm	<div></div>
HEAD /ping		487 ms	95 ms	18.73 rpm	<div></div>
GET /pour-rate		2,055 ms	145 ms	3.47 rpm	<div></div>
GET /brands/{id}([a-f0-9]{24})		4,859 ms	96 ms	0.9 rpm	<div></div>
GET /roast		20 ms	38 ms	211.2 rpm	<div></div>
GET /roast/{id}([a-f0-9]{24})		20 ms	38 ms	211.2 rpm	<div></div>



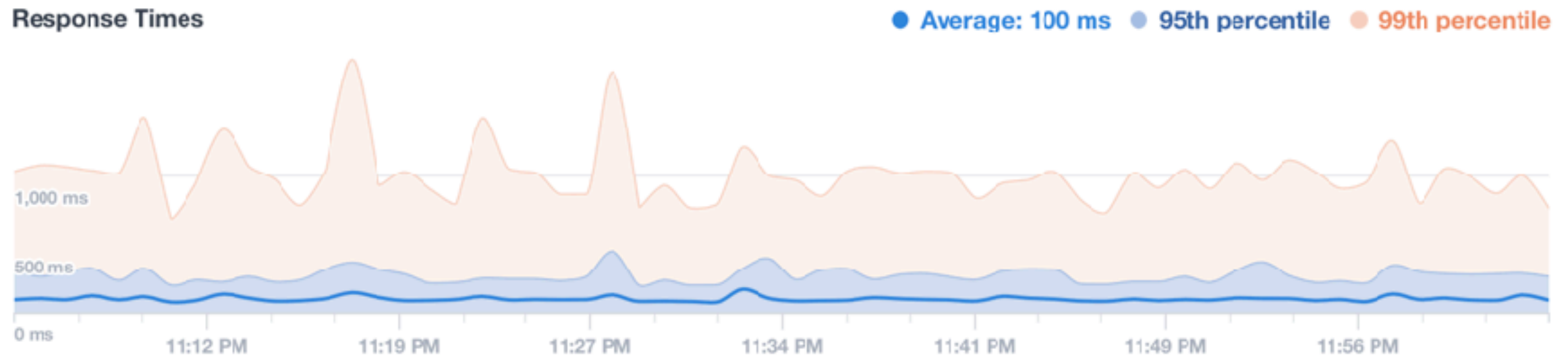
# Debugging Node.js in Production

Why is it important?

Why is it harder?

# Baseline performance

Response Times



Requests Per Minute



# Causes of Slowness

- (Single-threaded)
- CPU intensive code
- Slow I/O
- Event Loop saturation
- Running out of memory
- Garbage Collection



# CPU Intensive Code

- Synchronous I/O
- JSON.parse
- Regular Expressions
- Crypto
- Templates
- ...

How can we inspect what  
the CPU is doing for a  
given process?

# Sample Stack Traces

```
perf record -F 99 -p <pid> -g -- sleep 30
```

99 samples  
per second

Process to  
sample

Capture  
stack traces

Run for  
30 seconds

# Sample Stack Traces

```
perf record -F 99 -p <pid> -g -- sleep 30
```



perf.data



# Sample Stack Trace

```
751dcc v8::internal::Compiler::Compile (/usr/local/bin/node)
a84a96 v8::internal::Runtime_CompileLazy (/usr/local/bin/node)
223055f092a7 [unknown] (/tmp/perf-27806.map)
223055f34338 [unknown] (/tmp/perf-27806.map)
22305609a84f [unknown] (/tmp/perf-27806.map)
223056021453 [unknown] (/tmp/perf-27806.map)
223056020ebb [unknown] (/tmp/perf-27806.map)
223055f09895 [unknown] (/tmp/perf-27806.map)
223056098b9f [unknown] (/tmp/perf-27806.map)
22305609669f [unknown] (/tmp/perf-27806.map)
223055f3b7c3 [unknown] (/tmp/perf-27806.map)
223055f2508f [unknown] (/tmp/perf-27806.map)
84f994 v8::internal::Execution::Call (/usr/local/bin/node)
570a99 v8::Function::Call (/usr/local/bin/node)
57ec51 v8::Function::Call (/usr/local/bin/node)
c86a79 node::AsyncWrap::MakeCallback (/usr/local/bin/node)
cc9fd7 node::Parser::on_headers_complete (/usr/local/bin/node)
f0444b http_parser_execute (/usr/local/bin/node)
```

**JavaScript frames**

command: perf script

# Dump perf “meta data”

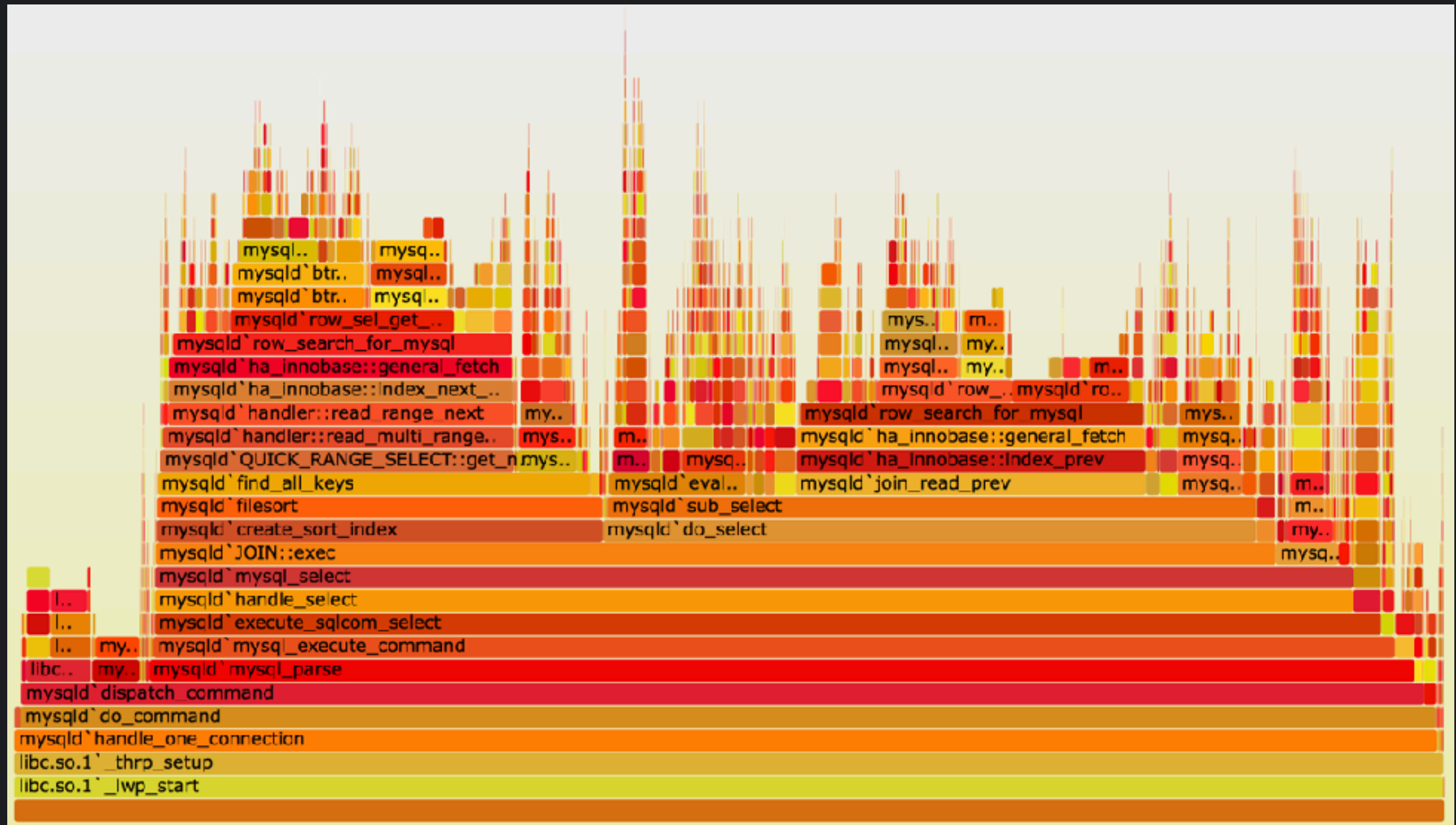
```
node --perf_basic_prof_only_functions server.js
```

# Sample Stack Trace with JavaScript frames

```
5e6f22 v8::internal::(anonymous namespace)::HandleApiCallHelper<false> (/usr/local/bin
5e759e v8::internal::Builtin_HandleApiCall (/usr/local/bin/node)
17712b1092a7 [unknown] (/tmp/perf-27742.map)
17712b29c49d LazyCompile:~pbkdf2 crypto.js:571 (/tmp/perf-27742.map)
17712b109895 [unknown] (/tmp/perf-27742.map)
17712b29c02f LazyCompile:~exports.pbkdf2Sync crypto.js:562 (/tmp/perf-27742.map)
17712b2ac721 LazyCompile:~foo /home/watson/1-cpu/cpu.js:13 (/tmp/perf-27742.map)
17712b29b4a3 LazyCompile:~ /home/watson/1-cpu/cpu.js:4 (/tmp/perf-27742.map)
17712b2221b3 LazyCompile:~emitTwo events.js:104 (/tmp/perf-27742.map)
17712b221cdb LazyCompile:~emit events.js:136 (/tmp/perf-27742.map)
17712b109895 [unknown] (/tmp/perf-27742.map)
17712b29979f LazyCompile:~parserOnIncoming _http_server.js:463 (/tmp/perf-27742.map)
17712b29729f LazyCompile:~parserOnHeadersComplete _http_common.js:45 (/tmp/perf-27742.map)
17712b13b7c3 [unknown] (/tmp/perf-27742.map)
17712b12508f [unknown] (/tmp/perf-27742.map)
84f994 v8::internal::Execution::Call (/usr/local/bin/node)
570a99 v8::Function::Call (/usr/local/bin/node)
57ec51 v8::Function::Call (/usr/local/bin/node)
c86a79 node::AsyncWrap::MakeCallback (/usr/local/bin/node)
cc9fd7 node::Parser::on_headers_complete (/usr/local/bin/node)
f0444b http_parser_execute (/usr/local/bin/node)
```

command: perf script

# Flame Graph





# Demo Time



# Postmortem Debugging

# mdb

- mdb (only runs on Solaris and friends)
- Solaris variant
- mdb v8 - [https://github.com/joyent/mdb\\_v8](https://github.com/joyent/mdb_v8)
- core dump (`--abort_on_uncaught_exception`)
- node binary

# Autopsy

```
JS Commands > ::dmods -l ! grep js  
V8 Commands > ::dmods -l ! grep v8  
Useful: ::status, ::dump, ::help <cmd>
```

<https://github.com/nearform/autopsy>



# Demo Time





Tak :)

@wa7son

[github.com/watson](https://github.com/watson)

 opbeat