sysdig

Universal System Visibility
With Native Container Support

Ramit Surana
@ramitsurana
/in/ramitsurana

Agenda

- Introduction to sysdig.
- Filtering
- Output Formatting
- Chisels
- Implementing Chisels
- Introducing Csysdig
- The Integrations
- Sysdig Conventions
- Sysdig Installation

sysdig

Who am I ?

- Open Source Tech Enthusiastic .
- Foodie,Traveler.
- Join me Here :
- Email:ramitsurana@gmail.com

  Twitter: @ramitsurana

  Linkedin: /in/ramitsurana

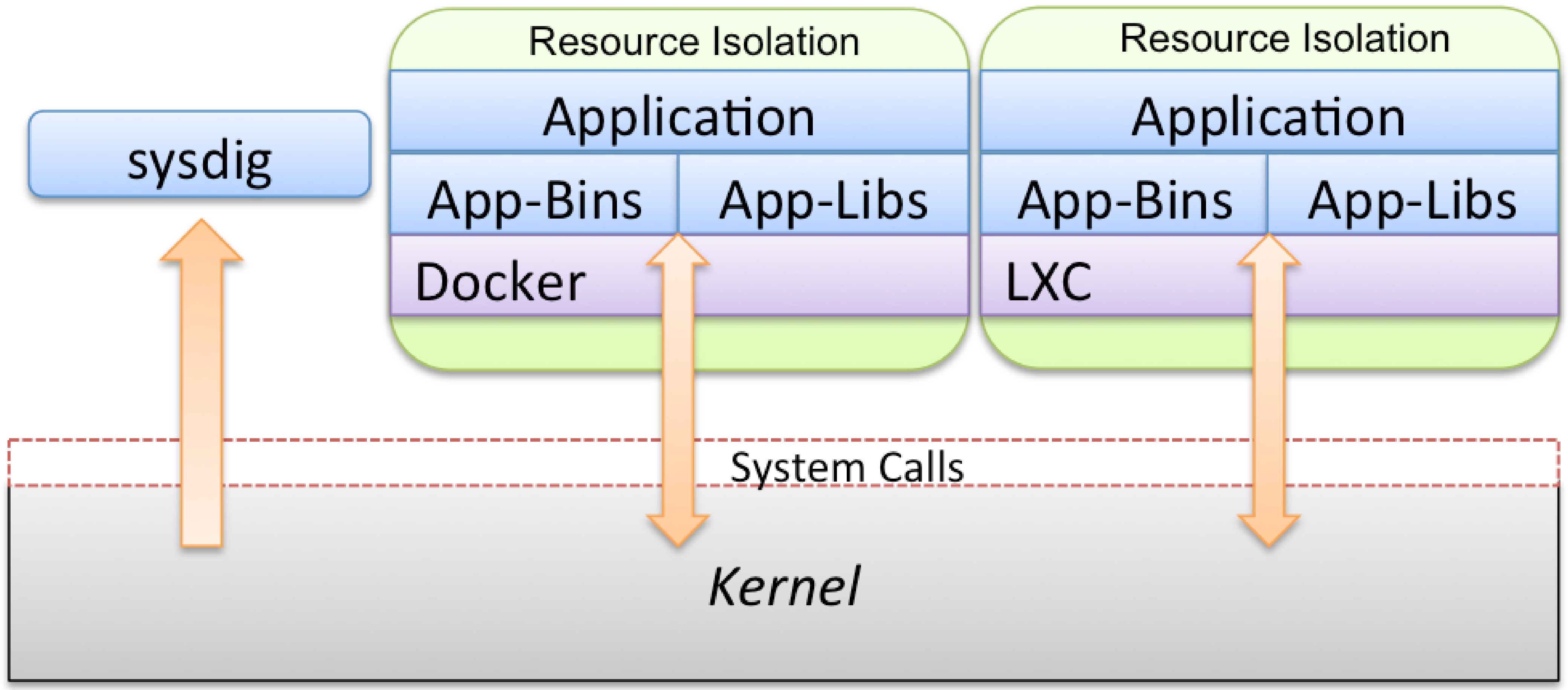
  Github:ramitsurana

sysdig

What is Sysdig ?

- New dynamic tracer for Linux, inspired by strace, dtrace, and tcpdump.
- In short: strace + tcpdump + htop + iftop + lsof + awesome sauce

Sysdig Architecture



sysdig

Hello Sysdig

- Some end of line arguments:
- evt.num is the incremental event number
- evt.time is the event timestamp
- evt.cpu is the CPU number where the event was captured
- proc.name is the name of the process that generated the event
- thread.tid is the TID that generated the event, which corresponds to the PID for single thread processes
- evt.dir is the event direction, > for enter events and < for exit events
- evt.type is the name of the event, e.g. 'open' or 'read'
- evt.args is the list of event arguments..

Filtering

- It is powerful and versatile, and is designed to look for needles in a haystack.
- Filters are specified at the end of the command line, like in tcpdump, and can be applied to both a live capture or a trace file.
- Filter statements can use the standard comparison operators(=, !=, <, <=, >, >=, contains) and can be combined using Boolean operators (and, or and not) and brackets.
- To list available filters: sysdig -l
- Some common filters & there usage:
- fd.name: To filter events for a specific file name
- proc.name: To capture all of the events for a specific process

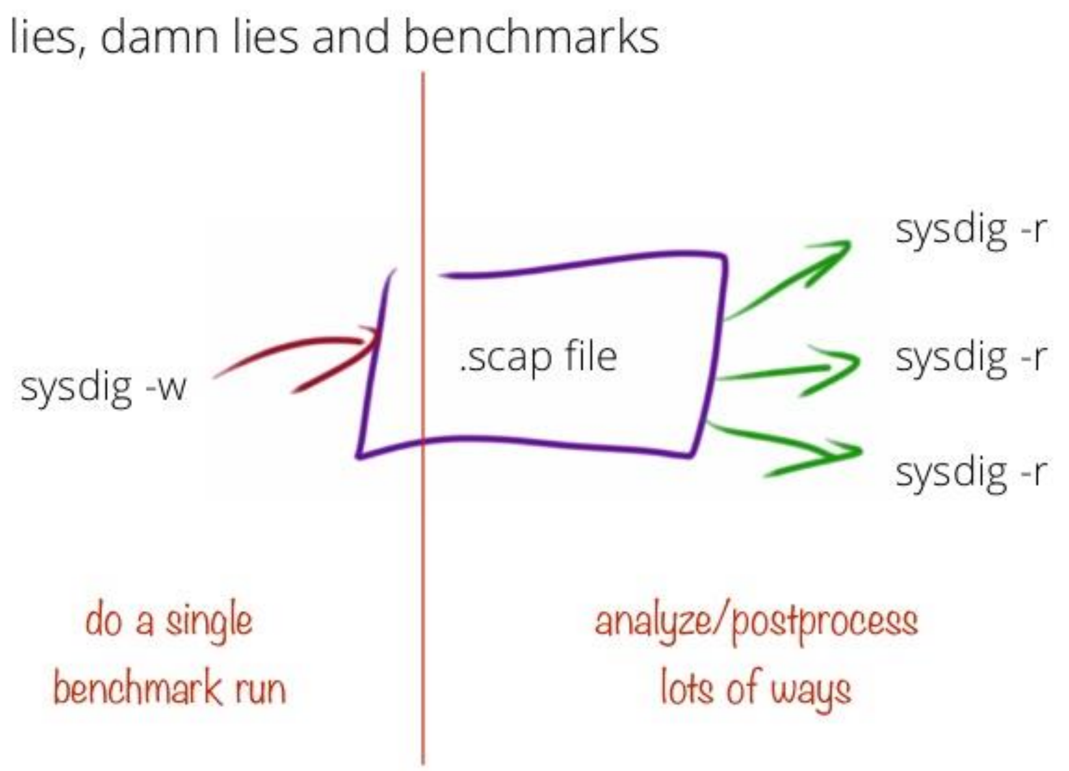
sysdig

Output Formatting

- It is powerful and versatile, it is designed to look for needles in a haystack.
- Filters are specified at the end of the command line, like in tcpdump, and can be applied to both a live capture or a trace file.
- Filter statements can use the standard comparison operators(=, !=, <, <=, >, >=, contains) and can be combined using Boolean operators (and, or and not) and brackets.

Chisels

- These are little scripts that analyze the sysdig event stream to perform useful actions.
- A well known scripting language can be used instead of a custom one. In fact, sysdig's chisels are Lua scripts. Lua is well known, powerful, stable and extremely efficient.
- Chisels can leverage the broad collection of Lua libraries.
- Chisels work well on live systems, but can also be used with trace files for offline analysis.


lies, damn lies and benchmarks
sysdig -w
.scap file
sysdig -r
sysdig -r
sysdig -r
do a single benchmark run
analyze/postprocess lots of ways

Implementing Chisels

- To run a chisel: sysdig -c <name of chisel>
- To display available chisels: sysdig -cl
- To give a small description of the chisels: sysdig -i <name of chisel>

Introducing Csysdig

- It exports sysdig's functionality through an intuitive and powerful ncurses-based user interface.
- It supports many features such as :
- Support for both live analysis and sysdig trace files. Trace files can come from the same machine or from another machine.
- Visibility into a broad range of metrics, including CPU, memory, disk I/O, network I/O.
- Ability to observe input/output activity for processes, files, network connections and more.
- Ability to drill down into processes, files, network connections and more to further explore their behavior.
- Support for sysdig's filtering language.
- Container support by design.

The Integrations

- Ansible
- Puppet Labs
- Elastic Search

And many more

Sysdig Covention

- Rules for committing code on Github in C++.
- Rules and instructions available at https://github.com/draios/sysdig/blob/master/coding_conventions.md.

Sysdig Installation

- Trust the Draios GPG key, configure the yum repository:

rpm --import https://s3.amazonaws.com/download.draios.com/DRAIOS-GPG-KEY.public

curl -s -o /etc/yum.repos.d/draios.repo http://download.draios.com/stable/rpm/draios.repo

- Install the EPEL repo:

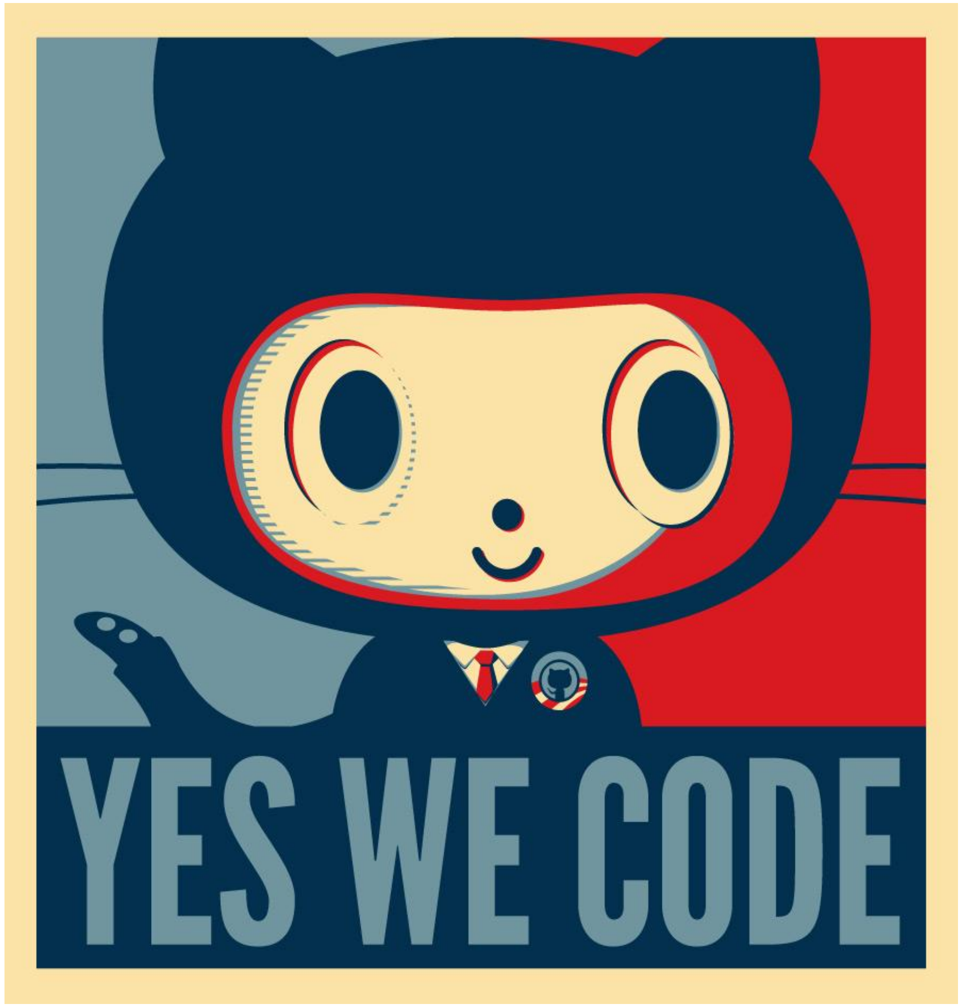
rpm -i http://mirror.us.leaseweb.net/epel/6/i386/epel-release-6-8.noarch.rpm

- Install the Kernel Header:

yum -y install kernel-devel-$(uname -r)

Questions ?

Please Contribute !!



Github.com/draios/sysdig

sysdig

Thank You

thanks for
listening!

Like it,Share it !!

sysdig