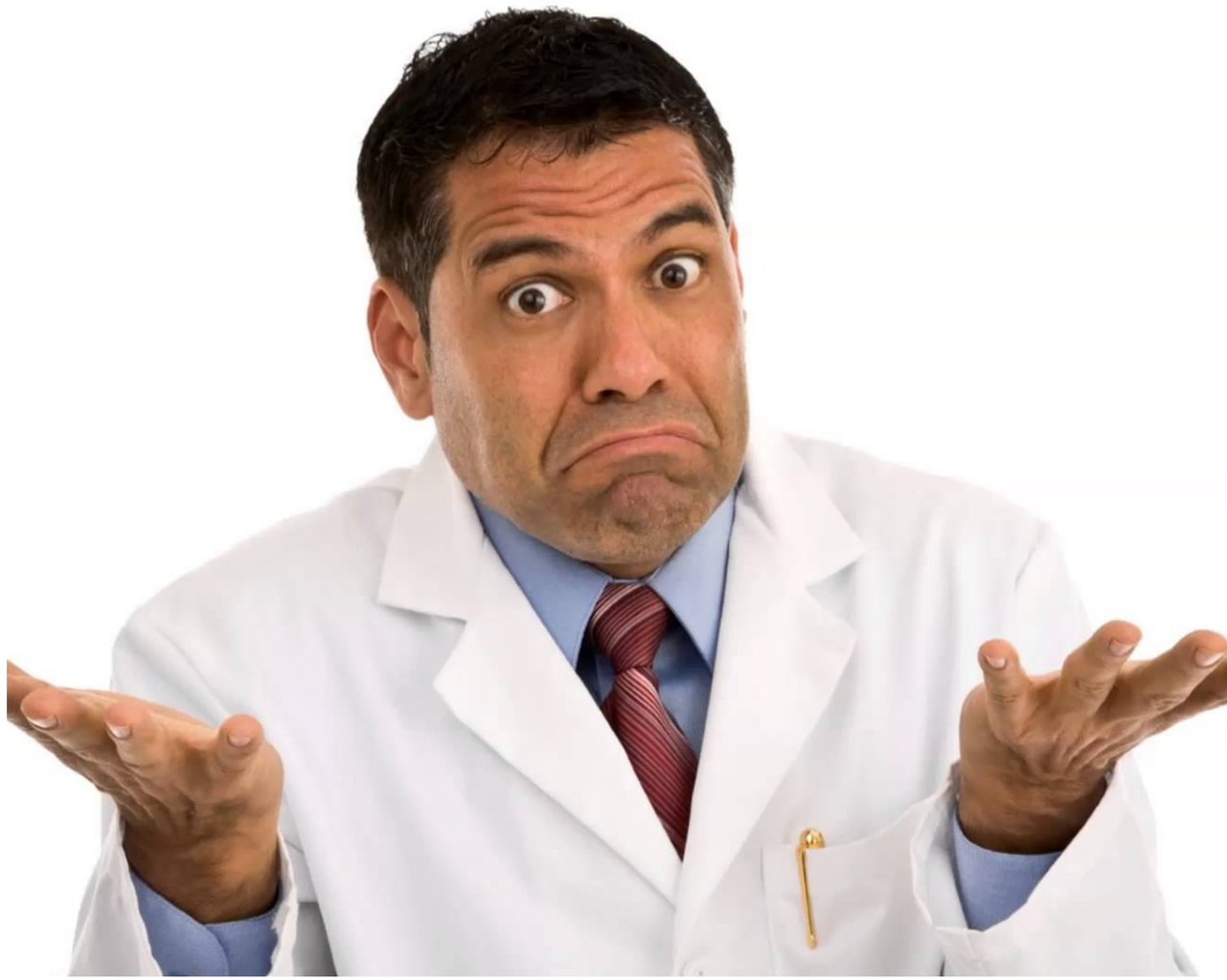
Monitoring Docker with Elastic Stack

Medcl, Elastic

Do you know if your containers are healthy?



elastic

Agenda

- What should you be monitoring in your microservice architecture?
- What is Metricbeat?
- How can Metricbeat help monitor you container infrastructure?

Built-in Tooling

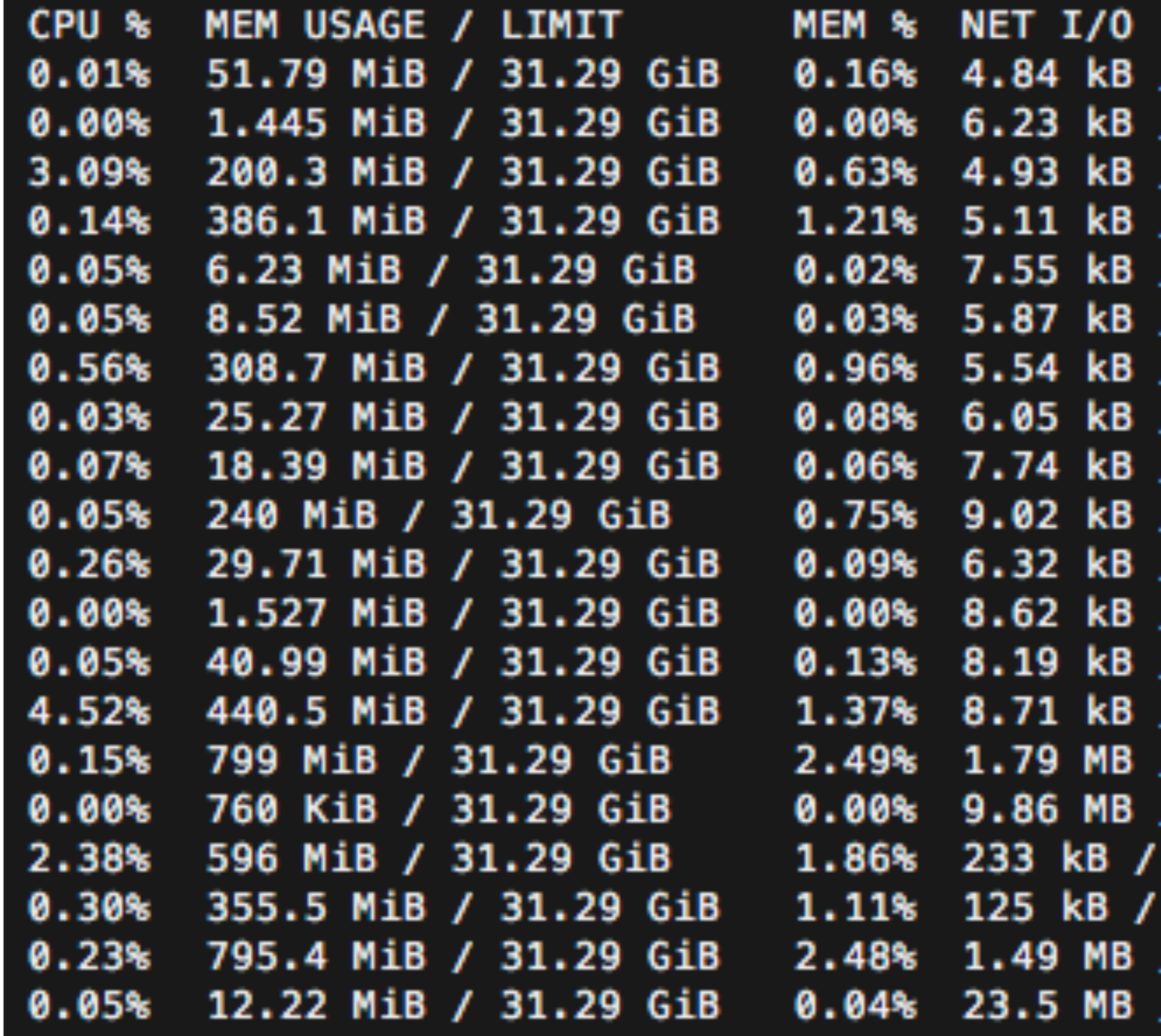
docker ps / docker stats

- Decentralized

No Historical Data

Reactive monitoring

Doesn’t Scale



CPU %	MEM USAGE / LIMIT	MEM %	NET I/O
0.01%	51.79 MiB / 31.29 GiB	0.16%	4.84 kB
0.00%	1.445 MiB / 31.29 GiB	0.00%	6.23 kB
3.09%	200.3 MiB / 31.29 GiB	0.63%	4.93 kB
0.14%	386.1 MiB / 31.29 GiB	1.21%	5.11 kB
0.05%	6.23 MiB / 31.29 GiB	0.02%	7.55 kB
0.05%	8.52 MiB / 31.29 GiB	0.03%	5.87 kB
0.56%	308.7 MiB / 31.29 GiB	0.96%	5.54 kB
0.03%	25.27 MiB / 31.29 GiB	0.08%	6.05 kB
0.07%	18.39 MiB / 31.29 GiB	0.06%	7.74 kB
0.05%	240 MiB / 31.29 GiB	0.75%	9.02 kB
0.26%	29.71 MiB / 31.29 GiB	0.09%	6.32 kB
0.00%	1.527 MiB / 31.29 GiB	0.00%	8.62 kB
0.05%	40.99 MiB / 31.29 GiB	0.13%	8.19 kB
4.52%	440.5 MiB / 31.29 GiB	1.37%	8.71 kB
0.15%	799 MiB / 31.29 GiB	2.49%	1.79 MB
0.00%	760 KiB / 31.29 GiB	0.00%	9.86 MB
2.38%	596 MiB / 31.29 GiB	1.86%	233 kB /
0.30%	355.5 MiB / 31.29 GiB	1.11%	125 kB /
0.23%	795.4 MiB / 31.29 GiB	2.48%	1.49 MB
0.05%	12.22 MiB / 31.29 GiB	0.04%	23.5 MB

Monitor all the things that are critical to the availability of your service.

1. Network
2. Hosts
3. Containers
4. Applications

Monitor all of the things that are critical to the availability of your service.

1. **Network** - switch / router monitoring via syslog and snmp traps with Logstash
2. Hosts
3. Containers
4. Applications

Monitor all of the things that are critical to the availability of your service.

1. Network
2. **Hosts** - Metricbeat
3. Containers
4. Applications

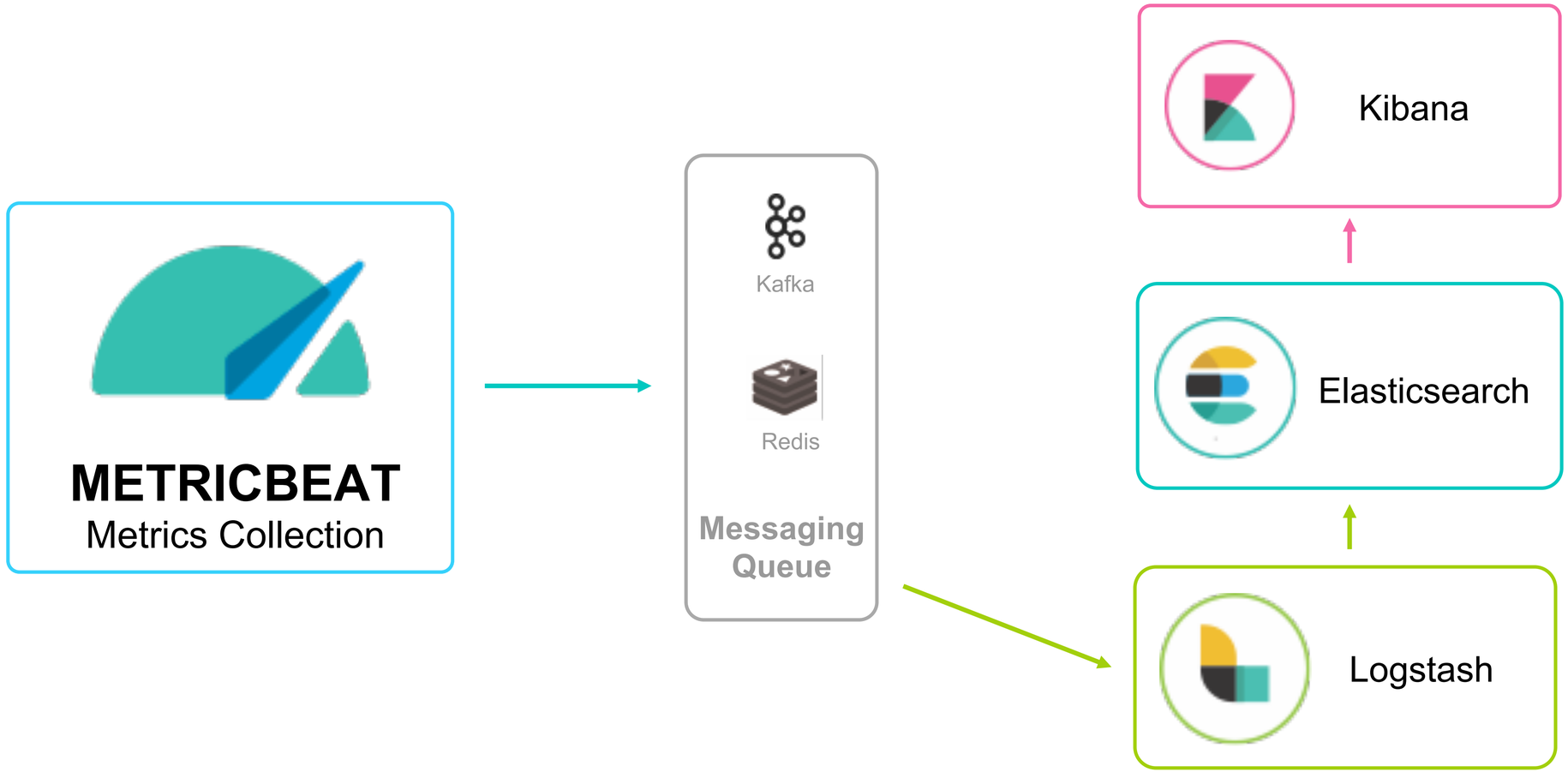
Use the Metricbeat system module to monitor your hosts.

```
metricbeat.modules:
- module: system
  metricsets:
    - cpu             # /proc/stat
    - load            # /proc/loadavg
    - memory          # /proc/meminfo
    - network         # /proc/net/dev
    - diskio          # /proc/diskstats
    - filesystem      # /etc/mtab -> statfs()
    - fsstat
    - process         # /proc/[pid]/{stat,status,statm,cmdline,environ}
  processes: ['.*']
  cgroups: true
```

Metricbeat can be deployed on the host or in a container.

```
docker run \
  -v /proc:/hostfs/proc:ro \
  -v /sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro \
  -v /:/hostfs:ro \
  -v /var/run/docker.sock:/var/run/docker.sock \
  my/metricbeat:latest -system.hostfs=/hostfs
```

Metricbeat has several outputs for architectural flexibility.



Elastic now has official container images for Beats.

```
FROM docker.elastic.co/beats/metricbeat:5.4.2

COPY metricbeat.yml /usr/share/metricbeat/metricbeat.yml
```

https://github.com/elastic/beats-docker

Monitor all the things that are critical to the availability of your service.

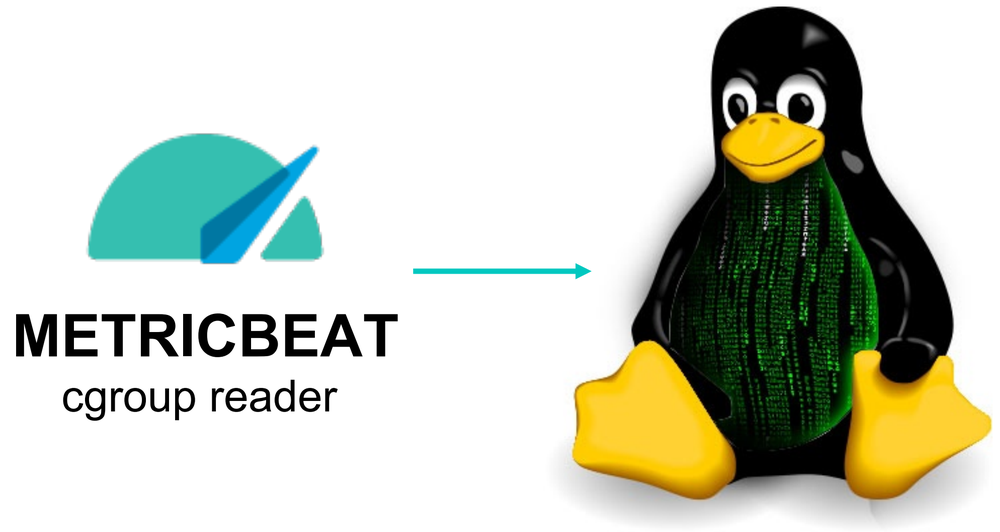
1. Network
2. Hosts
3. **Containers** - Metricbeat
4. Applications

Use the Metricbeat docker module to collect Docker daemon and container metrics.

```
metricbeat.modules:
- module: docker
  metricsets:
     - info              # daemon info
     - container         # id, image, command, labels, status
     - cpu               # kernel, user, total, per cpu usage
     - memory            # usage, max, limit
     - diskio            # disk io activity, reads, writes
     - network           # network io
     - healthcheck       # application health checked by a command
```

Metricbeat’s system process metricset collects cgroup metrics.

- Works with any container tool (Rocket, runC)
- No Docker API dependency
- Missing Docker metadata (name, labels)
- cgroup ID == container ID



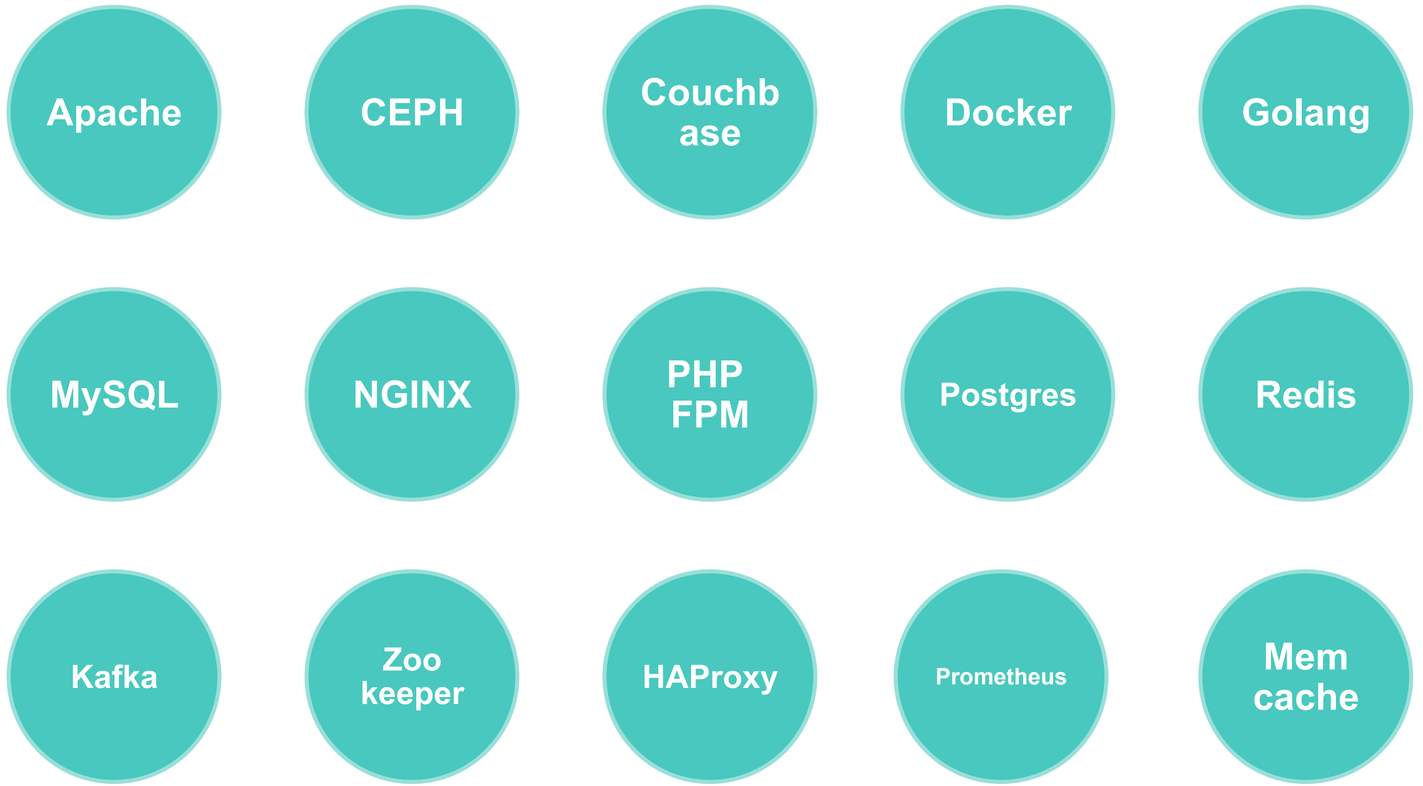
Monitor all of the things that are critical to the availability of your service.

1. Network
2. Hosts
3. Containers
4. **Applications** - Metricbeat

Use the Metricbeat modules to monitor common applications.

```
metricbeat.modules:
- module: mysql
  metricsets: [status]
  hosts: [“tcp(mysql01:3306)/"]
  username: root
  password: "${MYSQL_ROOT_PASSWORD}"
```

Monitor external services



Parse the Docker healthcheck output to get metrics from your custom apps.

In your Dockerfile:

```
HEALTHCHECK CMD curl -f -s http://localhost:8081/metrics?json
```

And in your metricbeat.yml:

```
processors:
- decode_json_fields:
    when.equals:
      docker.container.labels.service: myservice
    fields:
      - docker.healthcheck.event.output
    target: docker.healthcheck.myservice
```

Monitor all the things that are critical to the availability of your service.

1. Network
2. Hosts
3. Containers
4. Applications

Pro Tips

One Metricbeat container per Host

- Monitor the host, docker, and your apps from one container

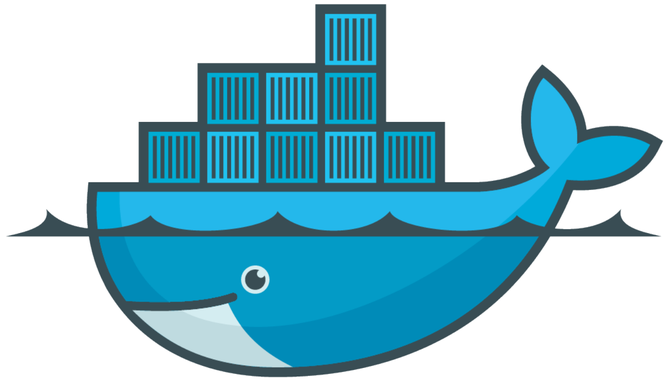
Use Labels

- Keep your sanity when scaling
- Makes it easier to build dashboards and alerts

Set Resource Limits

- Set bounds for CPU, memory, block I/O

Demo



DOCKER
Containers

METRICBEAT
Metrics

Dedicated Docker Registry

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:5.3.0

docker pull docker.elastic.co/logstash/logstash:5.3.0

docker pull docker.elastic.co/kibana/kibana:5.3.0
```

No "latest" tag.

No surprises.

https://flic.kr/p/sncAZ4

Built using functional and integration tests

```
test/common/test_base.py::test_base_os[docker://filebeat] PASSED
test/common/test_base.py::test_base_os[docker://heartbeat] PASSED
test/common/test_base.py::test_base_os[docker://metricbeat] PASSED
test/common/test_base.py::test_base_os[docker://packetbeat] PASSED
test/common/test_config.py::test_config_file_passes_config_test[docker://filebeat] PASSED
test/common/test_config.py::test_config_file_passes_config_test[docker://heartbeat] PASSED
test/common/test_config.py::test_config_file_passes_config_test[docker://metricbeat] PASSED
test/common/test_config.py::test_config_file_passes_config_test[docker://packetbeat] PASSED
test/common/test_files.py::test_binary_file_version[docker://filebeat] PASSED
test/common/test_files.py::test_binary_file_permissions[docker://filebeat] PASSED
test/common/test_files.py::test_binary_file_has_network_capabilities[docker://filebeat] PASSED
test/common/test_files.py::test_script_file_permissions[docker://filebeat] PASSED
test/common/test_files.py::test_config_file_permissions[docker://filebeat] PASSED
test/common/test_files.py::test_config_dir_permissions[docker://filebeat] PASSED
test/common/test_files.py::test_data_dir_permissions[docker://filebeat] PASSED
test/common/test_files.py::test_log_dir_permissions[docker://filebeat] PASSED
test/common/test_files.py::test_dashboard_archive_is_present[docker://filebeat] PASSED
test/common/test_files.py::test_template_locations[docker://filebeat] PASSED
test/common/test_files.py::test_binary_file_version[docker://heartbeat] PASSED
test/common/test_files.py::test_binary_file_permissions[docker://heartbeat] PASSED
test/common/test_files.py::test_binary_file_has_network_capabilities[docker://heartbeat] PASSED
test/common/test_files.py::test_script_file_permissions[docker://heartbeat] PASSED
test/common/test_files.py::test_config_file_permissions[docker://heartbeat] PASSED
test/common/test_files.py::test_config_dir_permissions[docker://heartbeat] PASSED
test/common/test_files.py::test_data_dir_permissions[docker://heartbeat] PASSED
test/common/test_files.py::test_log_dir_permissions[docker://heartbeat] PASSED
test/common/test_files.py::test_dashboard_archive_is_present[docker://heartbeat] PASSED
test/common/test_files.py::test_template_locations[docker://heartbeat] PASSED
test/common/test_files.py::test_binary_file_version[docker://metricbeat] PASSED
test/common/test_files.py::test_binary_file_permissions[docker://metricbeat] PASSED
test/common/test_files.py::test_binary_file_has_network_capabilities[docker://metricbeat] PASSED
test/common/test_files.py::test_script_file_permissions[docker://metricbeat] PASSED
test/common/test_files.py::test_config_file_permissions[docker://metricbeat] PASSED
test/common/test_files.py::test_config_dir_permissions[docker://metricbeat] PASSED
test/common/test_files.py::test_data_dir_permissions[docker://metricbeat] PASSED
test/common/test_files.py::test_log_dir_permissions[docker://metricbeat] PASSED
test/common/test_files.py::test_dashboard_archive_is_present[docker://metricbeat] PASSED
test/common/test_files.py::test_template_locations[docker://metricbeat] PASSED
test/common/test_files.py::test_binary_file_version[docker://packetbeat] PASSED
test/common/test_files.py::test_binary_file_permissions[docker://packetbeat] PASSED
test/common/test_files.py::test_binary_file_has_network_capabilities[docker://packetbeat] PASSED
test/common/test_files.py::test_script_file_permissions[docker://packetbeat] PASSED
test/common/test_files.py::test_config_file_permissions[docker://packetbeat] PASSED
test/common/test_files.py::test_config_dir_permissions[docker://packetbeat] PASSED
test/common/test_files.py::test_data_dir_permissions[docker://packetbeat] PASSED
test/common/test_files.py::test_log_dir_permissions[docker://packetbeat] PASSED
test/common/test_files.py::test_dashboard_archive_is_present[docker://packetbeat] PASSED
test/common/test_files.py::test_template_locations[docker://packetbeat] PASSED
test/common/test_process.py::test_process_is_pid_1[docker://filebeat] PASSED
test/common/test_process.py::test_process_is_running_as_the_correct_user[docker://filebeat] PASSED
test/common/test_process.py::test_process_was_started_with_the_foreground_flag[docker://filebeat] PASSED
test/common/test_process.py::test_process_is_pid_1[docker://heartbeat] PASSED
test/common/test_process.py::test_process_is_running_as_the_correct_user[docker://heartbeat] PASSED
test/common/test_process.py::test_process_was_started_with_the_foreground_flag[docker://heartbeat] PASSED
test/common/test_process.py::test_process_is_pid_1[docker://metricbeat] PASSED
test/common/test_process.py::test_process_is_running_as_the_correct_user[docker://metricbeat] PASSED
test/common/test_process.py::test_process_was_started_with_the_foreground_flag[docker://metricbeat] PASSED
test/common/test_process.py::test_process_is_pid_1[docker://packetbeat] PASSED
test/common/test_process.py::test_process_is_running_as_the_correct_user[docker://packetbeat] PASSED
test/common/test_process.py::test_process_was_started_with_the_foreground_flag[docker://packetbeat] PASSED
test/common/test_user.py::test_group_properties[docker://filebeat] PASSED
test/common/test_user.py::test_user_properties[docker://filebeat] PASSED
test/common/test_user.py::test_group_properties[docker://heartbeat] PASSED
test/common/test_user.py::test_user_properties[docker://heartbeat] PASSED
test/common/test_user.py::test_group_properties[docker://metricbeat] PASSED
```

Bundled with X-Pack

- Shield, Watcher, Marvel, Graph, reporting … soon also Machine Learning!
- Free Basic License provides monitoring and 18 zoom levels for the tile service

Different Ways to run the Elasticsearch image

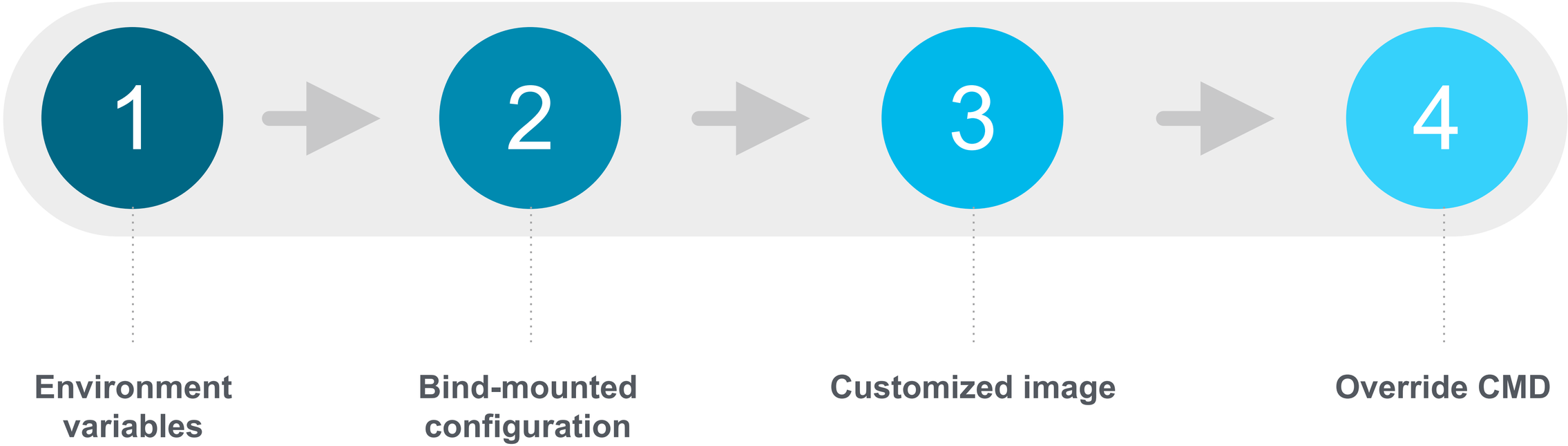
Dev Mode

- One line **docker run**
- Bypass bootstrap checks
- Do not run in production!

Prod Mode

- Default
- Bootstrap checks enabled
- Persistent volumes

Four ways to configure settings for Elasticsearch



Using Environment Variables for Elasticsearch

```
---
  elasticsearch:
    environment:
      - cluster.name=ESCluster
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms3g -Xmx3g"
```

Bind-mounted configuration

```
---
  elasticsearch:
    volumes:
      - "${PWD}/custom_esconfig.yml:\
        /usr/share/elasticsearch/config/elasticsearch.yml"
```

Make sure your local conf files can be read by Elasticsearch!

Build your own image!

```
FROM docker.elastic.co/elasticsearch/elasticsearch:5.2.1
ADD elasticsearch.yml /usr/share/elasticsearch/config/
```

chown **ADD**

elastic

Environment variables for Kibana

The same, but not the same!

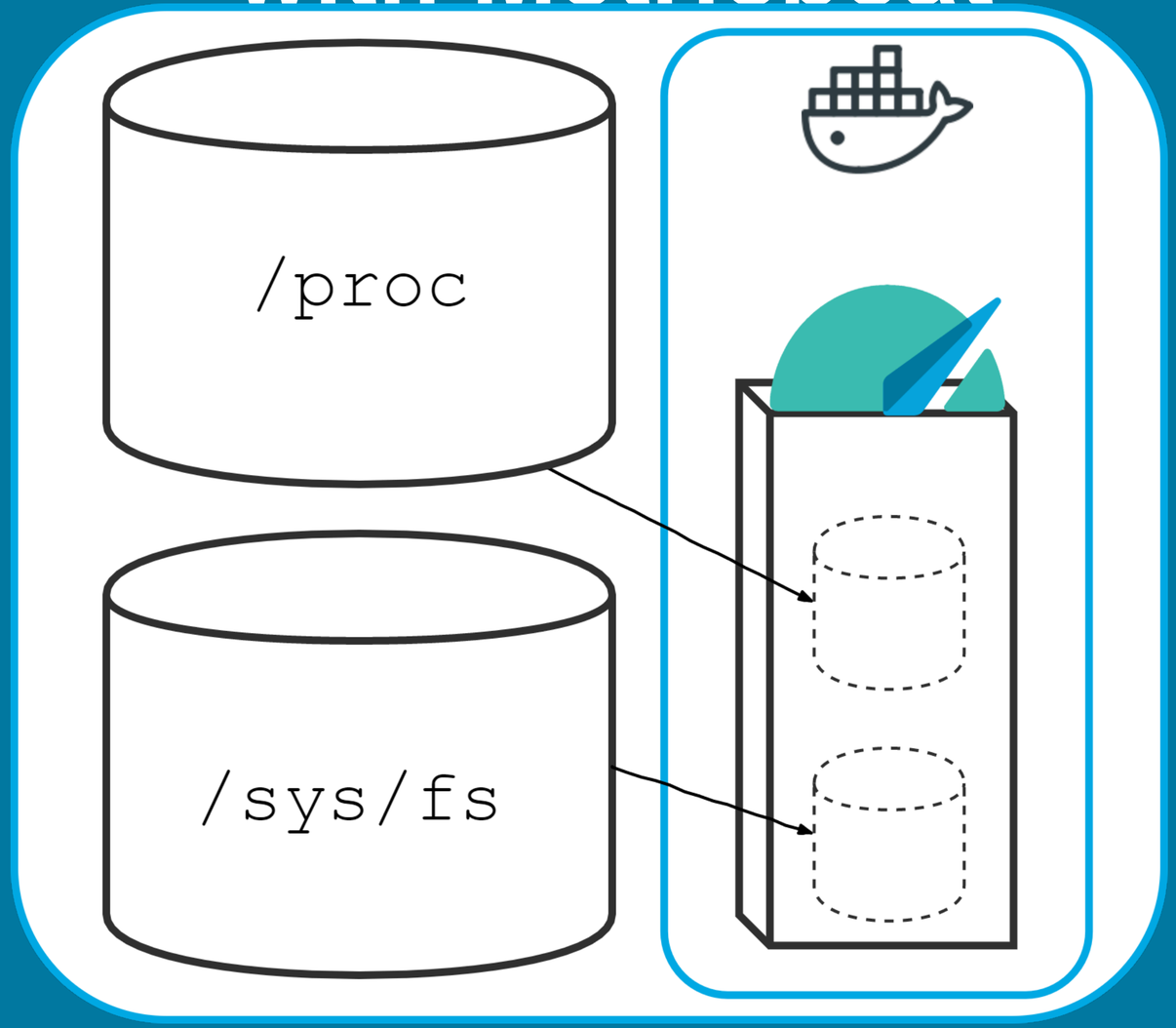
```
---
kibana:
  environment:
    - XPACK_SECURITY_ENABLED=true

elasticsearch:
  environment:
    - xpack.security.enabled=true
```

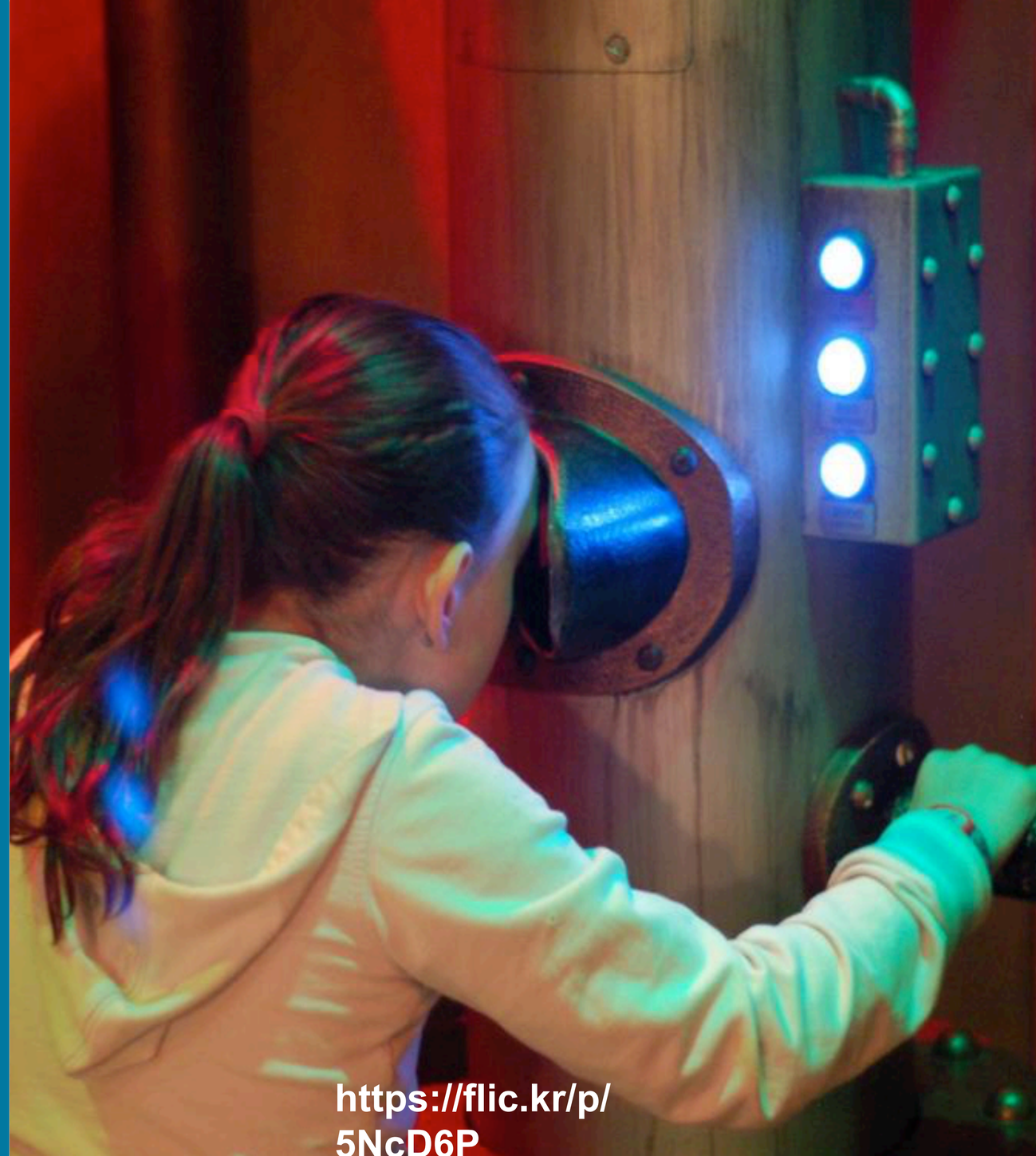
Logstash is a language

Ship your code.

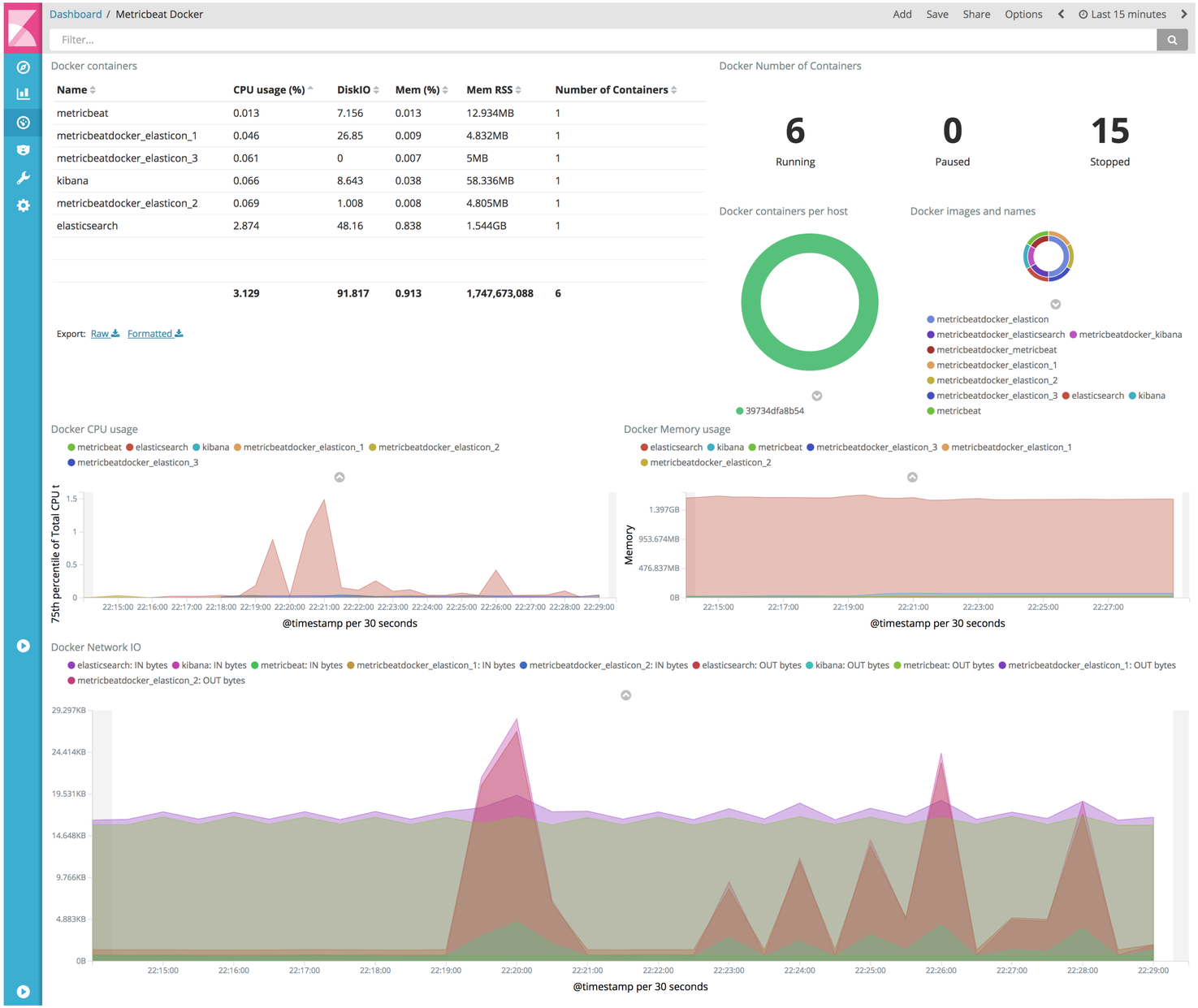
```
---
services:
  logstash:
    volumes:
      - ./config/:/usr/share/logstash/pipeline/
```

Host monitoring with Metricbeat





https://flic.kr/p/5NcD6P


Dashboard / Metricbeat Docker
Add
Save
Share
Options
Last 15 minutes
Filter...
Docker containers
Name
CPU usage (%)
DiskIO
Mem (%)
Mem RSS
Number of Containers
metricbeat 0.013 7.156 0.013 12.934MB 1
metricbeatdocker_elasticon_1 0.046 26.85 0.009 4.832MB 1
metricbeatdocker_elasticon_3 0.061 0 0.007 5MB 1
kibana 0.066 8.643 0.038 58.336MB 1
metricbeatdocker_elasticon_2 0.069 1.008 0.008 4.805MB 1
elasticsearch 2.874 48.16 0.838 1.544GB 1
3.129 91.817 0.913 1,747,673,088 6
Export: Raw Formatted
Docker Number of Containers
6
Running
0
Paused
15
Stopped
Docker containers per host
39734dfa8b54
Docker images and names
metricbeatdocker_elasticon
metricbeatdocker_elasticsearch
metricbeatdocker_kibana
metricbeatdocker_metricbeat
metricbeatdocker_elasticon_1
metricbeatdocker_elasticon_2
metricbeatdocker_elasticon_3
elasticsearch
kibana
metricbeat
Docker CPU usage
metricbeat
elasticsearch
kibana
metricbeatdocker_elasticon_1
metricbeatdocker_elasticon_2
metricbeatdocker_elasticon_3
75th percentile of Total CPU t
1.5
1
0.5
0
22:15:00 22:16:00 22:17:00 22:18:00 22:19:00 22:20:00 22:21:00 22:22:00 22:23:00 22:24:00 22:25:00 22:26:00 22:27:00 22:28:00 22:29:00
@timestamp per 30 seconds
Docker Memory usage
elasticsearch
kibana
metricbeat
metricbeatdocker_elasticon_3
metricbeatdocker_elasticon_1
metricbeatdocker_elasticon_2
Memory
1.397GB
953.674MB
476.837MB
0B
22:15:00 22:17:00 22:19:00 22:21:00 22:23:00 22:25:00 22:27:00
@timestamp per 30 seconds
Docker Network IO
elasticsearch: IN bytes
kibana: IN bytes
metricbeat: IN bytes
metricbeatdocker_elasticon_1: IN bytes
metricbeatdocker_elasticon_2: IN bytes
elasticsearch: OUT bytes
kibana: OUT bytes
metricbeat: OUT bytes
metricbeatdocker_elasticon_1: OUT bytes
metricbeatdocker_elasticon_2: OUT bytes
29.297KB
24.414KB
19.531KB
14.648KB
9.766KB
4.883KB
0B
22:15:00 22:16:00 22:17:00 22:18:00 22:19:00 22:20:00 22:21:00 22:22:00 22:23:00 22:24:00 22:25:00 22:26:00 22:27:00 22:28:00 22:29:00
@timestamp per 30 seconds

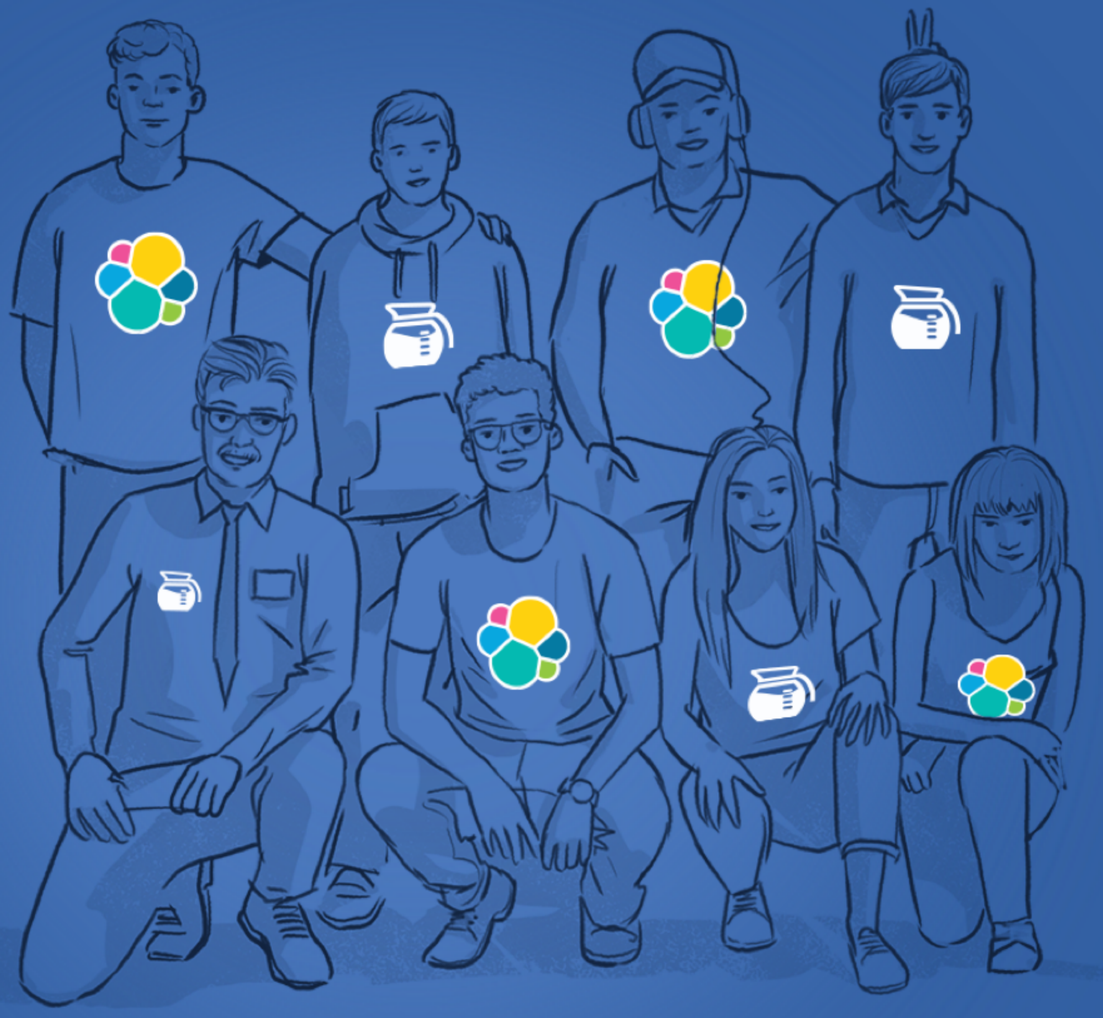
DEMO

opbeat
an Elastic company

React AngularJS Node.js Pricing Docs Community Log in Sign up

Opbeat is joining forces with Elastic.

Read our blog post



elastic

www.elastic.co

END


Dashboard / Elastic{ON} Docker Demo
Add
Save
Share
Options
Last 15 minutes
docker.container.labels.co_elastic_service_name:elasticon
Elastic{ON} Demo App Metrics
300
250
200
150
100
50
0
22:37
22:38
22:39
22:40
22:41
22:42
22:43
22:44
22:45
22:46
22:47
22:48
22:49
22:50
22:51
200 272
404 272
Total Requests 272
Elastic{ON} Demo App Container CPU
metricbeatdocker_elasticon_1 0.01
metricbeatdocker_elasticon_3 0.01
metricbeatdocker_elasticon_2 0.01
Elastic{ON} Demo App Top Servers
metricbeatdocker_elasticon_1 33,099
metricbeatdocker_elasticon_3 33,097
metricbeatdocker_elasticon_2 33,096
Elastic{ON} Demo App Container Memory
metricbeatdocker_elasticon_1 17.0MB
metricbeatdocker_elasticon_2 14.3MB
metricbeatdocker_elasticon_3 13.5MB
Average CPU Usage
1.3%
Average Mem Usage
14.9MB

Sample healthcheck event


```
"docker": {
  "container": {
    "id": "83c3355dc06f8b2fbe225c4e3638bd025f5140ef638c6b115e85995c217fbe1f",
    "labels": {
      "co_elastic_service_name": "elasticon",
      "com_docker_compose_config-hash": "ee66c7f94c1a134acad038ac9bc583424806f96446f594f623c45c76f615ff4e",
      "com_docker_compose_container-number": "1",
      "com_docker_compose_oneoff": "False",
      "com_docker_compose_project": "metricbeatdocker",
      "com_docker_compose_service": "elasticon",
      "com_docker_compose_version": "1.11.2"
    },
    "name": "metricbeatdocker_elasticon_1"
  },
  "healthcheck": {
    "elasticon": {
      "http_200": 15032,
      "http_404": 20875,
      "http_total_requests": 35907
    },
    "event": {
      "end_date": "2017-03-08T07:06:51.514Z",
      "exit_code": 0,
      "output": "{\n\"http_200\": 15032,\n\"http_404\": 20875,\n\"http_total_requests\": 35907\n}\n",
      "start_date": "2017-03-08T07:06:51.345Z"
    },
    "failingstreak": 0,
    "status": "healthy"
  }
}
```