

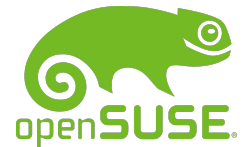
# openSUSE® 2020

## Monitoring and managing Containers using Open Source tools

Monitoring and management with sysdig, portainer and  
Rancher



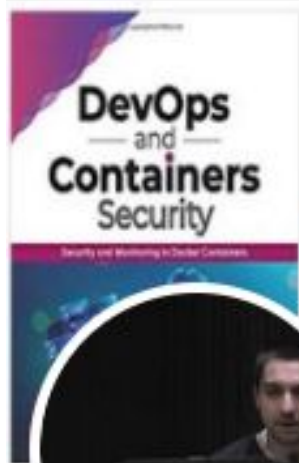
Jose Manuel Ortega  
@jmortegac



@jmortegac

<http://jmortega.github.io>

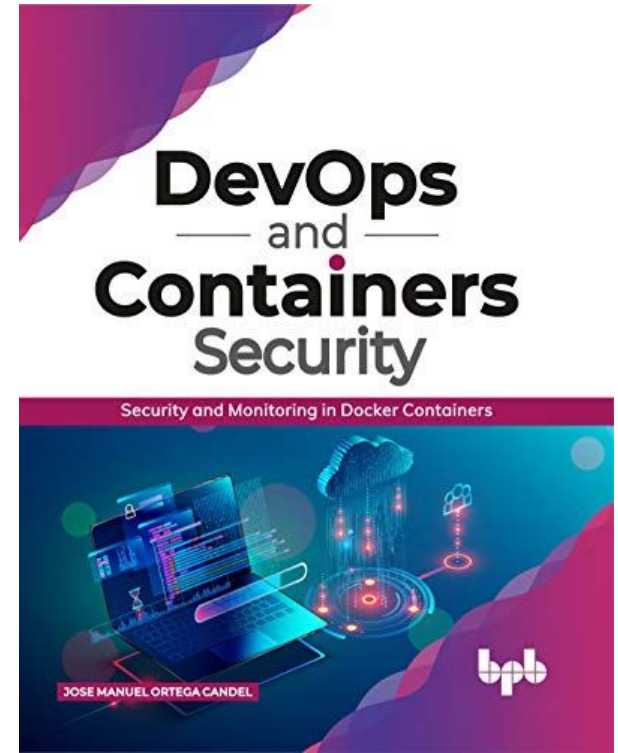
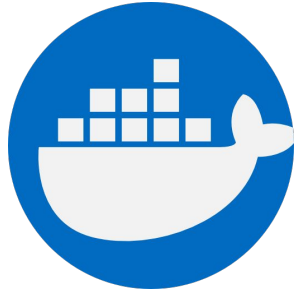
<https://www.linkedin.com/in/jmortega1/>



**J. M. Ortega**

@jmortegac

Editar perfil



<https://bpbonline.com/products/devops-and-containers-security-security-and-monitoring-in-docker-containers>



- **Challenges** in containers and architectures distributed.
- **Metrics** that we can use to measure container performance.
- Tools for monitoring and management of containers such as **cadvisor**, **sysdig** and **portainer**.
- **Rancher** as a platform for the administration of Kubernetes.

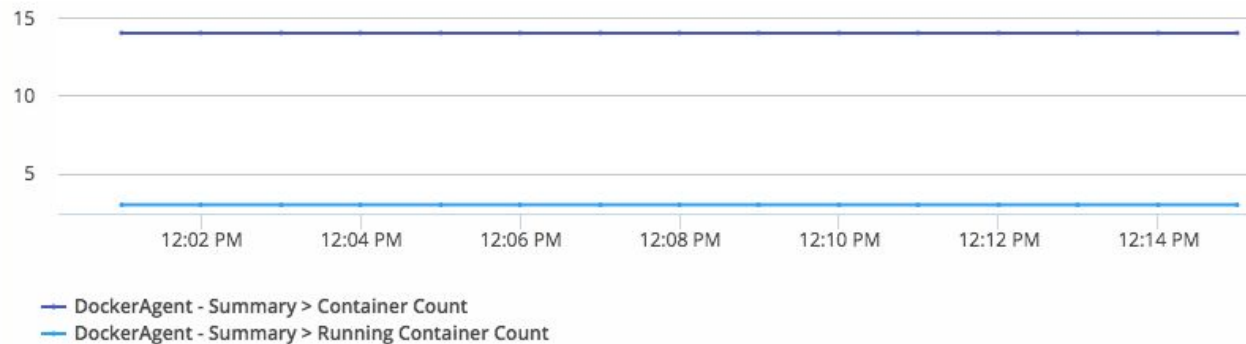


- Detecting issues and troubles.
- **Changes and upgrades can be done safely.**
- Refine applications for better performance and robustness.



docker

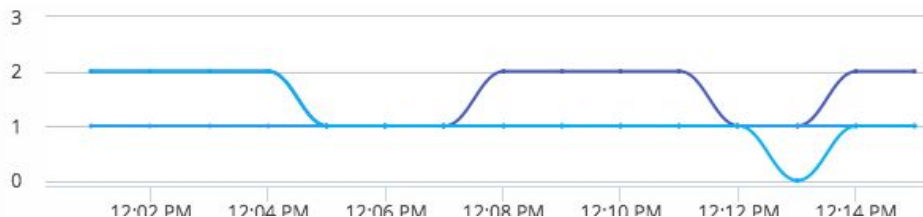
Total vs Running Containers



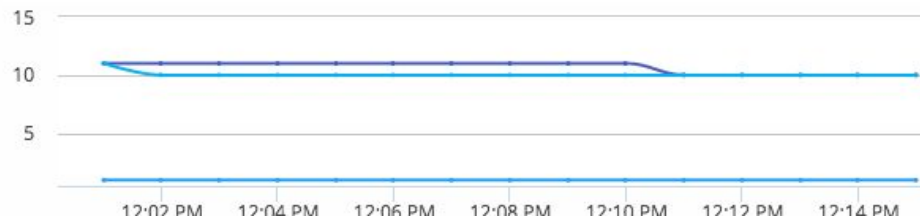
Total Containers - 14

Total Images - 192

Total CPU %



Memory Usage %



# Finding the Most Relevant Metrics



**CPU**



**Input/Output**



**Memory**



**Network  
traffic**

# Docker stats

```
[root@docker ~]# docker stats
```

CONTAINER	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
fb874de145f2	0.00%	480 KiB / 3.788 GiB	0.01%	1.3 kB / 648 B	0 B / 0 B	0
258adf2f71fb	0.00%	7.375 MiB / 3.788 GiB	0.19%	4.07 kB / 648 B	9.38 MB / 807 kB	0
9254e01fadad	0.00%	3.445 MiB / 3.788 GiB	0.09%	5.36 kB / 648 B	3.67 MB / 0 B	0
465ca90475d8	0.00%	72.53 MiB / 3.788 GiB	1.87%	1.63 kB / 1.43 kB	69.5 MB / 0 B	0
e24e59379bd7	0.01%	206.1 MiB / 3.788 GiB	5.31%	2.08 kB / 1.24 kB	34.1 MB / 25.5 MB	0
9149cc6dfb35	0.11%	96.95 MiB / 1 GiB	9.47%	99.9 MB / 96.1 MB	116 MB / 43.3 MB	0
0e18349d6651	0.04%	523.9 MiB / 1 GiB	51.17%	88.9 MB / 99.1 MB	45.5 MB / 3.07 GB	0
128f71c54c52	0.00%	13.43 MiB / 256 MiB	5.25%	2.72 kB / 648 B	22.1 MB / 0 B	0
CONTAINER	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
fb874de145f2	0.00%	480 KiB / 3.788 GiB	0.01%	1.3 kB / 648 B	0 B / 0 B	0
258adf2f71fb	0.00%	7.375 MiB / 3.788 GiB	0.19%	4.07 kB / 648 B	9.38 MB / 807 kB	0
9254e01fadad	0.00%	3.445 MiB / 3.788 GiB	0.09%	5.36 kB / 648 B	3.67 MB / 0 B	0
465ca90475d8	0.01%	72.53 MiB / 3.788 GiB	1.87%	1.63 kB / 1.43 kB	69.5 MB / 0 B	0
e24e59379bd7	0.11%	206.1 MiB / 3.788 GiB	5.31%	2.08 kB / 1.24 kB	34.1 MB / 25.5 MB	0
9149cc6dfb35	0.01%	96.95 MiB / 1 GiB	9.47%	99.9 MB / 96.1 MB	116 MB / 43.3 MB	0
0e18349d6651	0.09%	523.9 MiB / 1 GiB	51.17%	88.9 MB / 99.1 MB	45.5 MB / 3.07 GB	0
128f71c54c52	0.00%	13.43 MiB / 256 MiB	5.25%	2.72 kB / 648 B	22.1 MB / 0 B	0



# docker stats [OPTIONS] [CONTAINER...]

```
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT  MEM %               NET I/O
      BLOCK I/O          PIDS
84e51c306399       some-mysql2        96.37%             418MiB / 31.4GiB  1.30%              0B / 0B
      4.1kB / 413MB        43
215608e92f7f       some-mysql         0.34%              385MiB / 31.4GiB  1.20%              0B / 0B
      0B / 494MB           38
```

# Docker Remote API

**`/containers/{id}/stats`**

**`/containers/{name}/stats`**

# curl -s [http://localhost:2375/v1.12/containers](http://localhost:2375/v1.12/containers/<id_container>/stats) /<id\_container>/stats

```
$ curl -s http://localhost:2375/v1.12/containers/84e51c306399/stats
{"read":"2019-06-04T18:55:47.404233989Z","preread":"0001-01-01T00:00:00Z","pids_stats":{"current":37},"blkio_stats":{"io_service_bytes_recursive":[{"major":8,"minor":16,"op":"Read","value":4096}, {"major":8,"minor":16,"op":"Write","value":496304128}, {"major":8,"minor":16,"op":"Sync","value":496304128}, {"major":8,"minor":16,"op":"Async","value":4096}, {"major":8,"minor":16,"op":"Total","value":496308224}], "io_serviced_recursive":[{"major":8,"minor":16,"op":"Read","value":1}, {"major":8,"minor":16,"op":"Write","value":11583}, {"major":8,"minor":16,"op":"Sync","value":11583}, {"major":8,"minor":16,"op":"Async","value":1}, {"major":8,"minor":16,"op":"Total","value":11584}], "io_queue_recursive":[], "io_service_time_recursive":[], "io_wait_time_recursive":[], "io_merged_recursive":[], "io_time_recursive":[], "sectors_recursive":[]}, "num_procs":0, "storage_stats":{}, "cpu_stats":{"cpu_usage":{"total_usage":13172500755, "percpu_usage":[1546083907, 1293089818, 1436136768, 337035061, 1393323222, 1340916822, 3008723263, 2817191894], "usage_in_kernelmode":4410000000, "usage_in_usermode":7070000000}, "system_cpu_usage":9535706360000000, "online_cpus":8, "throttling_data":{"periods":0, "throttled_periods":0, "throttled_time":0}}, "precpu_stats":{"cpu_usage":{"total_usage":0, "usage_in_kernelmode":0, "usage_in_usermode":0}, "throttling_data":{"periods":0, "throttled_periods":0, "throttled_time":0}}, "memory_stats":{"usage":587800576, "max_usage":638910464, "stats":{"active_anon":387895296, "active_file":70082560, "cache":183566336, "dirty":0, "hierarchical
```

cA d v i s o r



<https://github.com/google/cadvisor>



cAdvisor

```
docker run \  
  --volume=:/rootfs:ro \  
  --volume=/var/run:/var/run:rw \  
  --volume=/sys:/sys:ro \  
  --volume=/var/lib/docker:/var/lib/docker:ro \  
  --publish=8080:8080 \  
  --detach=true \  
  --name=cadvisor \  
  google/cadvisor:latest
```

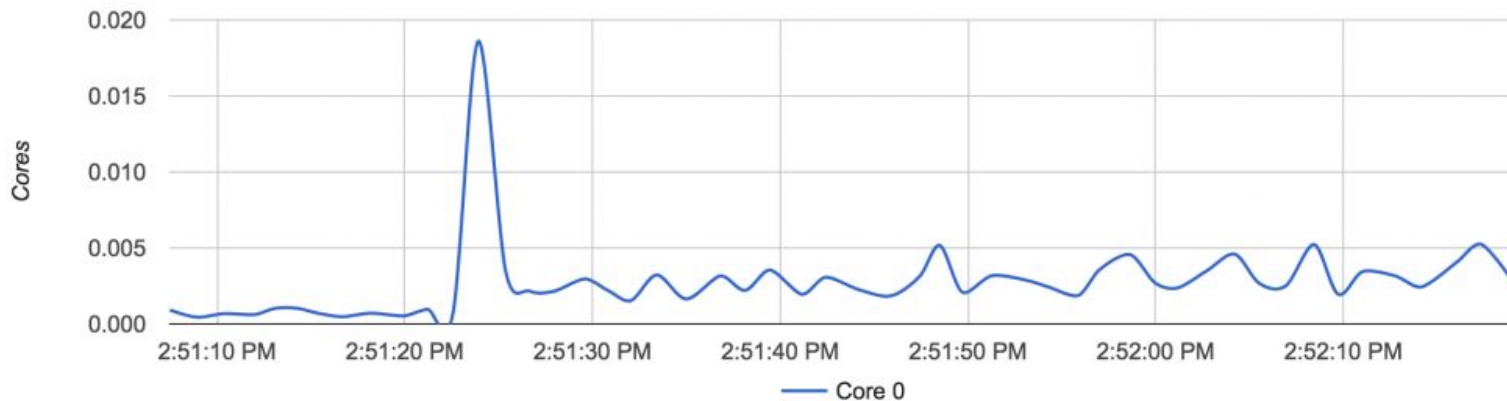
## Overview



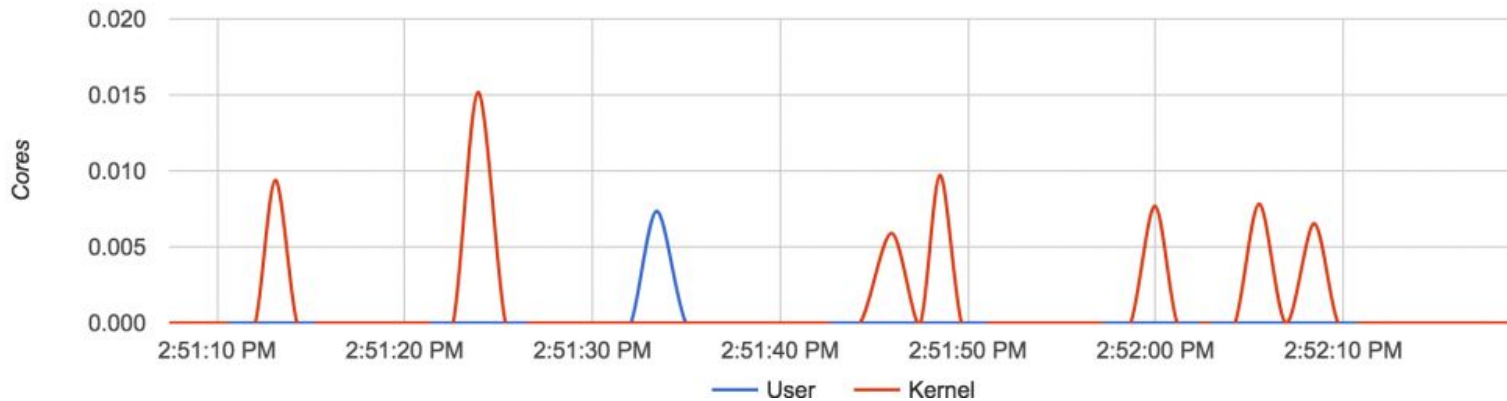
## Processes

User	PID	PPID	Start Time	CPU % ▼	MEM %	RSS	Virtual Size	Status	Running Time	Command	Container
root	257	1	12:57	1.90	10.10	75.01 MiB	346.67 MiB	Ss1	00:00:06	dockerd	/system.slice/docker.service
root	924	908	13:02	1.70	4.40	32.81 MiB	718.91 MiB	Ss1	00:00:01	cadvisor	/docker/0d31fb6cc13298b27f6bf7
root	1	0	12:57	0.20	1.10	8.34 MiB	82.98 MiB	Ss	00:00:00	systemd	/
root	278	257	12:57	0.10	2.10	16.15 MiB	205.18 MiB	Ss1	00:00:00	docker-containe	/system.slice/docker.service
root	2	0	12:57	0.00	0.00	0.00 B	0.00 B	S	00:00:00	kthreadd	/
root	4	2	12:57	0.00	0.00	0.00 B	0.00 B	I<	00:00:00	kworker/0:0:H	/
root	6	2	12:57	0.00	0.00	0.00 B	0.00 B	I<	00:00:00	mm_percpu_wq	/
root	7	2	12:57	0.00	0.00	0.00 B	0.00 B	S	00:00:00	ksoftirqd/0	/
root	8	2	12:57	0.00	0.00	0.00 B	0.00 B	I	00:00:00	rcu_sched	/
root	9	2	12:57	0.00	0.00	0.00 B	0.00 B	I	00:00:00	rcu_bh	/
root	10	2	12:57	0.00	0.00	0.00 B	0.00 B	S	00:00:00	migration/0	/
root	11	2	12:57	0.00	0.00	0.00 B	0.00 B	S	00:00:00	watchdog/0	/
root	12	2	12:57	0.00	0.00	0.00 B	0.00 B	S	00:00:00	cruid/0	/

## Usage per Core



## Usage Breakdown



cAdvisor



Prometheus



# Docker Containers

[Docker Containers](#)

## Subcontainers

[portainer.1.r2ie59dycqkl8fi9vg25audw9 \(/docker/4f37a0893442872ea4bf3d856583ae5b9588ffa1850f3dbce38e8379f2f019f7\)](#)

[cadvisor \(/docker/0d31fb6cc13298b27f6bf72653057a355bc9f1b1104bdcb2c54417c2780c1eee\)](#)

## Driver Status

**Docker Version** 18.03.1-ce

**Docker API Version** 1.37

**Kernel Version** 4.14.67-1-lts

**OS Version** Arch Linux





# Others Docker container monitoring tool

- <https://ctop.sh>

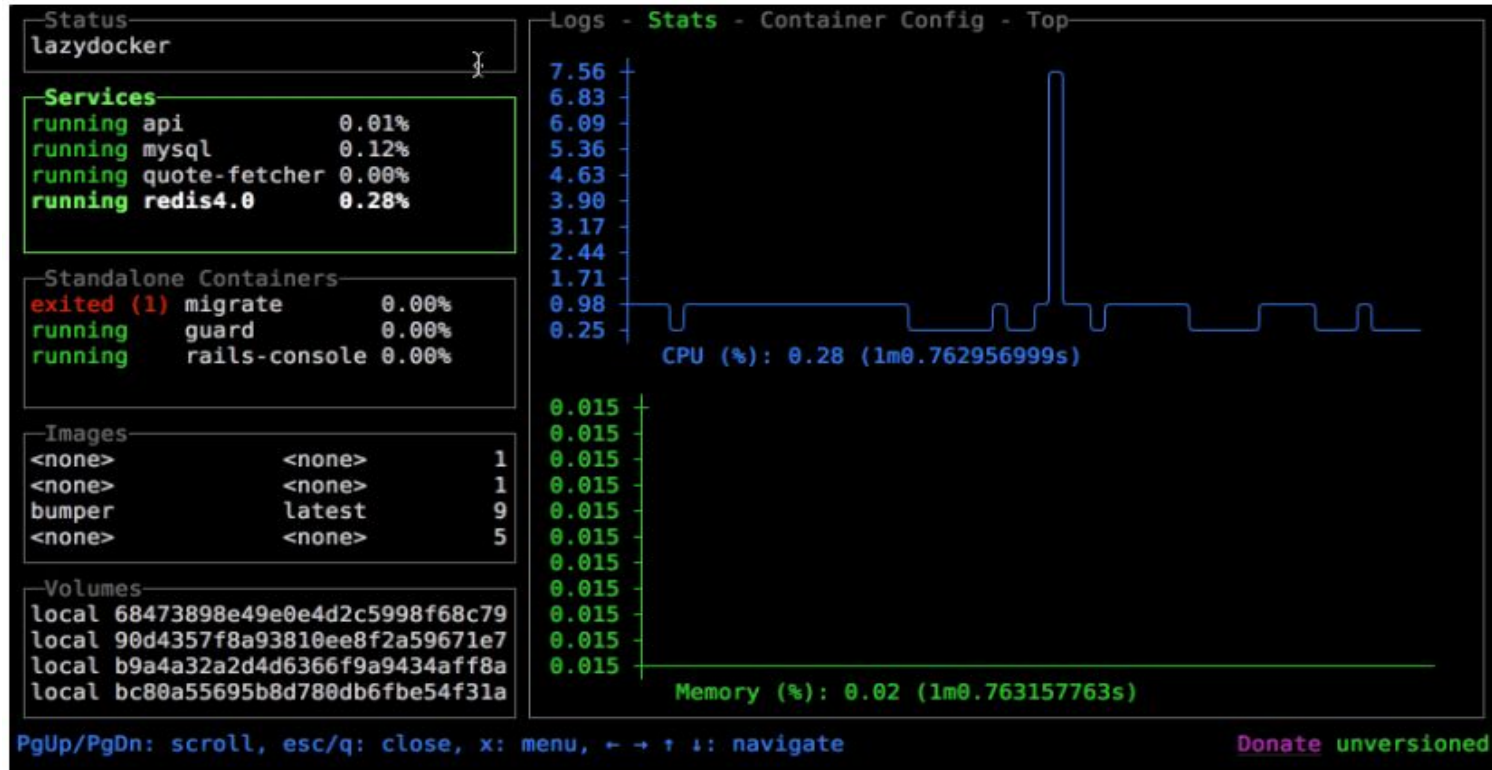
```
cTop - 10:04:12 AEDT      20 containers

NAME                CID                CPU                MEM                NET RX/TX
● luminous_lady      cf754eb3aa09      39%                125M / 2G         2K / 1K
● ultimate_jennifer 9eb1e9a6cb91      36%                121M / 2G         1K / 2K
● top_notch          7b5fd634a980      33%                153M / 2G         2K / 2K
● neat_roulette      0677c5437698      17%                53M / 2G          598B / 619B
● exquisite_jackpot 72372078a320      16%                53M / 2G          680B / 694B
● ace_void           292db4fa7c5c      13%                55M / 2G          661B / 674B
● neat_multiple      5fbe5928be03      13%                44M / 2G          874B / 779B
● legendary_korath   23a40fbb1cb       10%                51M / 2G          751B / 743B
● peachy_sakura      70bd0664ff8c      10%                46M / 2G          671B / 672B
● astonishing_nikita 93de62c2b03f      -                  -                  -
● cats_pajamas       ec19826bd862      -                  -                  -
● fantabulous_titane... 8e23ebb7e05f      -                  -                  -
● grand_bebop        c3a231ea8f49      -                  -                  -
■ impressive_sentine... e154df51f8e6      -                  -                  -
```



# Others Docker container monitoring tool

- <https://github.com/jesseduffield/lazydocker>





- Exploring layers in docker images
  - <https://github.com/wagoodman/dive>

```
1: Terminal ▾
[● Layers]
Cmp Image ID      Size Command
sha256:cd7100a72410606589 4.1 MB FROM sha256:cd7100a72410606589
sha256:f03b1ccbacace8c82e 2.1 kB #(nop) ADD file:63d4894bd0857354b
sha256:d2a26525cdac6c4358 0 B mkdir /root/example
sha256:9f50561518484bb62a 2.1 kB cp /somefile.txt /root/example/so
sha256:edf209d09263ab03c7 2.1 kB cp /somefile.txt /root/example/so
sha256:bb70c7aab83602e8a6 2.1 kB cp /somefile.txt /root/example/so
sha256:b0a712d3d00bc83c48 2.1 kB mv /root/example/somefile3.txt /r
sha256:05e8660b8f90e77b93 0 B rm -rf /root/example/

[Image & Layer Details]
Layer Command
/bin/sh -c mv /root/example/somefile3.txt /root/saved.txt

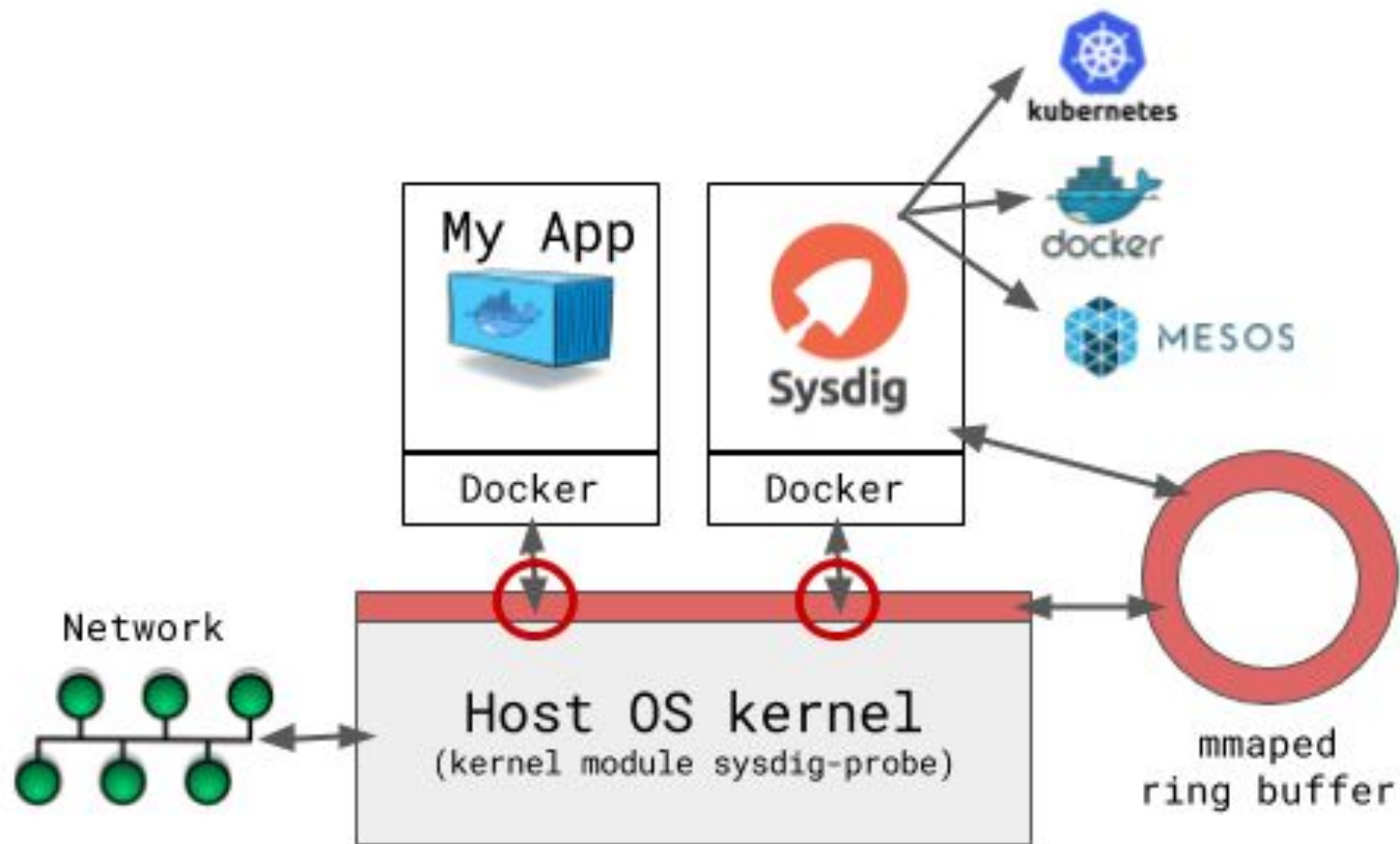
Image efficiency score: 99 %
Potential wasted space: 6.2 kB

Count Total Space Path
2 4.2 kB /root/example
2 2.1 kB /root/example/somefile3.txt

[Current Layer Contents]
Permission UID:GID Size Filetree
drwxr-xr-x 0:0 805 kB bin
drwxr-xr-x 0:0 0 B dev
drwxr-xr-x 0:0 251 kB etc
drwxr-xr-x 0:0 0 B home
drwxr-xr-x 0:0 2.7 MB lib
drwxr-xr-x 0:0 0 B media
drwxr-xr-x 0:0 0 B cdrom
drwxr-xr-x 0:0 0 B floppy
drwxr-xr-x 0:0 0 B usb
drwxr-xr-x 0:0 0 B mnt
drwxr-xr-x 0:0 0 B proc
drwx----- 0:0 6.2 kB root
drwxr-xr-x 0:0 4.2 kB example
-rw-r--r-- 0:0 2.1 kB somefile1.txt
-rw-r--r-- 0:0 2.1 kB somefile2.txt
-rw-r--r-- 0:0 2.1 kB somefile3.txt
-rw-r--r-- 0:0 2.1 kB saved.txt
drwxr-xr-x 0:0 0 B run
drwxr-xr-x 0:0 213 kB sbin
-rw-rw-r-- 0:0 2.1 kB somefile.txt
drwxr-xr-x 0:0 0 B srv
drwxr-xr-x 0:0 0 B sys
drwxrwxrwx 0:0 0 B tmp
drwxr-xr-x 0:0 176 kB usr
drwxr-xr-x 0:0 0 B var
```



**sysdig**





Source: Live System Filter: evt.type!=switch

PID	CPU	USER	TH	VIRT	RES	FILE	NET	Command
2545	0.50		32	2G	217M	0	0.00	mysqld
6254	0.50	root	1	83M	14M	0	0.00	csysdig
2704	0.00	root	10	147M	2M	0	0.00	docker-containerd-shim 0ed238220f8295664f
b9e 1	0.00	root	1	33M	3M	0	0.00	/sbin/init
3311	0.00	www-data	1	312M	29M	0	0.00	apache2 -DFOREGROUND
1071	0.00	root	1	14M	916K	0	0.00	/sbin/getty -8 38400 tty6
930	0.00	root	1	42M	2M	0	0.00	/lib/systemd/systemd-logind
2599	0.00	root	9	138M	2M	0	0.00	docker-containerd-shim b17e05614381d91974
903309	0.00	www-data	1	311M	27M	0	0.00	apache2 -DFOREGROUND
1069	0.00	root	1	14M	928K	0	0.00	/sbin/getty -8 38400 tty3
3320	0.00		8	212M	9M	5K	2.62K	docker run -it --rm --name=sysdig --privi
1108	0.00	root	1	23M	868K	0	0.00	cron
3306	0.00	www-data	1	387M	38M	0	0.00	apache2 -DFOREGROUND
2832	0.00	root	8	11M	3M	0	0.00	forego start -r
2924	0.00	_apt	1	14M	2M	0	0.00	nginx: worker process
878	0.00	root	19	501M	32M	0	2.90K	/usr/bin/dockerd -H tcp://0.0.0.0:2345 -H
u3381	0.00	root	8	258M	4M	5K	0.00	docker-containerd-shim e35a422ea2a6412b2d
ce1157	0.00		1	17M	800K	0	2.00K	/usr/bin/dirmngr --daemon --sh
1060	0.00	root	1	14M	928K	0	0.00	/sbin/getty -8 38400 tty4
963	0.00	root	14	253M	10M	0	138.50	docker-containerd -l unix:///var/run/dock
er2813	0.00	root	10	211M	2M	0	0.00	docker-containerd-shim e0d30f9aef18d210bd

F1Help F2Views F4Filter F5Echo F6Dig F7Legend F8Actions F9Sort F12Spectro CTRL+F Searchp Pau

12/61(19)

# Sysdig falco

<https://sysdig.com/opensource/falco/>



```
root@e51c99888dd2:/# sysdig -L
> syscall(SYSCALLID ID, UINT16 nativeID)
< syscall(SYSCALLID ID)
> open()
< open(FD fd, FSPATH name, FLAGS32 flags, UINT32 mode)
> close(FD fd)
< close(ERRNO res)
> read(FD fd, UINT32 size)
< read(ERRNO res, BYTEBUF data)
> write(FD fd, UINT32 size)
< write(ERRNO res, BYTEBUF data)
> socket(FLAGS32 domain, UINT32 type, UINT32 proto)
< socket(FD fd)
> bind(FD fd)
< bind(ERRNO res, SOCKADDR addr)
> connect(FD fd)
< connect(ERRNO res, SOCKTUPLE tuple)
> listen(FD fd, UINT32 backlog)
< listen(ERRNO res)
> send(FD fd, UINT32 size)
< send(ERRNO res, BYTEBUF data)
```

```
root@e51c99888dd2:/# sysdig -cl
```

Category: Application

```
-----
httplog           HTTP requests log
httptop          Top HTTP requests
memcachelog      memcached requests log
```

Category: CPU Usage

```
-----
spectrogram      Visualize OS latency in real time.
subsecoffset     Visualize subsecond offset execution time.
topcontainers_cpu
topprocs_cpu     Top containers by CPU usage
                 Top processes by CPU usage
```

# Sysdig falco

<https://sysdig.com/opensource/falco/>



```
root@e51c99888dd2:/# sysdig -c lscontainers
container.type container.image container.name container.id
-----
docker sysdig/sysdig sysdig e51c99888dd2
docker jwilder/nginx-p proxy a2d0e31c5dff
docker wordpress wp2 8534b21035f9
docker wordpress wp1 6e7eb5a84835
docker mysql mysql 64ff677ad564
```



# Sysdig falco

<https://sysdig.com/opensource/falco/>



```
root@e51c99888dd2:/# sysdig -cl

Category: Application
-----
httplog      HTTP requests log
httpstop     Top HTTP requests
memcachelog  memcached requests log

Category: CPU Usage
-----
spectrogram  Visualize OS latency in real time.
subsecoffset Visualize subsecond offset execution time.
topcontainers_cpu
              Top containers by CPU usage
topprocs_cpu  Top processes by CPU usage

Category: Errors
-----
topcontainers_error
              Top containers by number of errors
topfiles_errors Top files by number of errors
topprocs_errors top processes by number of errors
```

# Sysdig falco

<https://sysdig.com/opensource/falco/>



```
root@e35a422ea2a6:/# sysdig -pc -c httplog
sysdig -pc -c httplog
2018-01-26 16:20:23.621328421 host < method=GET url=unix.sock/_ping response_code=200 latency=0ms size=2B
2018-01-26 16:20:33.622424143 host < method=GET url=unix.sock/_ping response_code=200 latency=0ms size=2B
2018-01-26 16:20:33.622465554 proxy > method=GET url=unix.sock/_ping response_code=200 latency=10001ms size=2B
```

```
root@e35a422ea2a6:/# sysdig -v -A -s 2048 "(proc.name=apache2 or proc.name=mysql) and evt.type!=gettimeofd
and evt.type!=switch and evt.type!=io_getevents and evt.type!=futex and evt.type!=clock_gettime and evt.typ
epoll_wait and evt.type!=getsockopt and evt.type!=wait4 and evt.type!=select and evt.type!=semop"
615 16:29:15.078285566 0 mysql (2760) < nanosleep res=0
616 16:29:15.078308436 0 mysql (2760) > nanosleep interval=1000000000(1s)
2146 16:29:16.078403520 0 mysql (2760) < nanosleep res=0
2147 16:29:16.078424031 0 mysql (2760) > nanosleep interval=1000000000(1s)
3617 16:29:17.078523735 0 mysql (2760) < nanosleep res=0
3618 16:29:17.078546042 0 mysql (2760) > nanosleep interval=1000000000(1s)
5592 16:29:18.078639448 0 mysql (2760) < nanosleep res=0
5593 16:29:18.078660758 0 mysql (2760) > nanosleep interval=1000000000(1s)
7495 16:29:19.078767107 0 mysql (2760) < nanosleep res=0
7496 16:29:19.078790839 0 mysql (2760) > nanosleep interval=1000000000(1s)
9119 16:29:20.078856583 0 mysql (2760) < nanosleep res=0
9120 16:29:20.078876279 0 mysql (2760) > nanosleep interval=1000000000(1s)
```





## Add Environment

Name

Kubernetes

Description

e.g. Environment for developer experimentation

### Environment Template



**Orchestration:** Kubernetes

**Framework:** Healthcheck Service, Network Services, Scheduler

**Networking:** Rancher IPsec









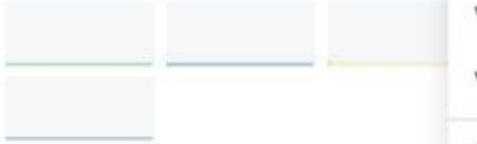





**Containers**
[Add Container](#)
 Show System






Actions ▾

 Search:
 

&lt;&lt; &lt; 25 Items &gt; &gt;&gt;

<input type="checkbox"/>	State ▾	Name ▾	IP Address ▾	Host ▾	Image ▾	Command ▾	
<input type="checkbox"/>	Starting	coredns	None	master	sha256:eb516548c180f8a6e023	None	<input type="checkbox"/>
<input type="checkbox"/>	Starting	coredns	None	master	sha256:eb516548c180f8a6e023	None	<input type="checkbox"/>
<input type="checkbox"/>	Stopped	coredns-fb8b8dccb-ndxix	10.42.226.169	master	k8s.gcr.io/pause:3.1	None	<input type="checkbox"/>
<input type="checkbox"/>	Stopped	coredns-fb8b8dccb-zcf7x	10.42.7.135	master	k8s.gcr.io/pause:3.1	None	<input type="checkbox"/>
<input type="checkbox"/>	Stopped	etcd	None	master	sha256:2c4adeb21b4ff8ed3309c	None	<input type="checkbox"/>
<input type="checkbox"/>	Stopped	etcd-master	None	master	k8s.gcr.io/pause:3.1	None	<input type="checkbox"/>
<input type="checkbox"/>	Running	healthcheck-healthcheck-1	10.42.76.44	master	rancher/healthcheck:v0.3.8	healthcheck,--metadata-addr...	<input type="checkbox"/>
<input type="checkbox"/>	Running	ipsec cni driver 1	None	master	rancher/net:v0.13.17	start cni driver.sh	<input type="checkbox"/>
<input type="checkbox"/>	Running	ipsec-ipsec-1	10.42.135.94	master	rancher/net:holder	sh.-c.echo Refer to router side...	<input type="checkbox"/>

<input type="radio"/> Running	nginx	10.42.29.199	nginx		 
<input type="radio"/> Running	rancher-agent	None	rancher/agent:v1.2.1:		 
<input type="radio"/> Running	scheduler-schedu...	10.42.114.193	rancher/scheduler:v0		 
<input type="radio"/> Running	unruffled_neuma...	172.18.0.2	rancher/server:stable		 

- Restart 
- Stop 
- Delete 
- Execute Shell
- View Logs
- View in API 
- Clone 
- Edit 



portainer.io

HOME

SOFTWARE & SERVICES ▾

OVERVIEW

# MAKING DOCKER MANAGEMENT EASY.

Build and manage your Docker environments with ease today.

GET IT NOW

LIVE DEMO

Available on  LINUX,  WINDOWS, &  OSX





## portainer/portainer ☆

By [portainer](#) • Updated 19 days ago

Making Docker management easy. <https://portainer.io>

Container

↓ Pulls **1B+**

Overview

Tags

docker pulls **2.1G** **0B** **6 layers** docs **passing** chat on gitter Donate PayPal

**Portainer** is a lightweight management UI which allows you to **easily** manage your Docker host or Swarm cluster.

**Portainer** is meant to be as **simple** to deploy as it is to use. It consists of a single container that can run on any Docker engine (Docker for Linux and Docker for Windows are supported)

### Docker Pull Command

```
docker pull portainer/portainer
```

```
docker run -d -p 9000:9000 --name portainer --restart always -v  
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer
```

```
version: '3'  
volumes:  
  portainer_data:  
services:  
  portainer:  
    image: portainer/portainer  
    container_name: portainer  
    restart: unless-stopped  
    volumes:  
      - portainer_data:/data  
      - /var/run/docker.sock:/var/run/docker.sock  
    ports:  
      - 9000:9000
```

Connect Portainer to the Docker environment you want to manage.

 **Local**

Manage the local Docker environment



 **Remote**

Manage a remote Docker environment

 **Agent**

Connect to a Portainer agent

 **Azure**

Connect to Microsoft Azure ACI

## Information

Connect Portainer to a remote Docker environment using the Docker API over TCP.

 The Docker API must be exposed over TCP. You can find more information about how to expose the Docker API over TCP [in the Docker documentation](#).

## Environment

**Name**

e.g. docker-prod01

**Endpoint URL** 

e.g. 10.0.0.10:2375 or mydocker.mydomain.com:2375

**TLS** 



 **Connect**



Home



PRIMARY

Dashboard



App Templates



Stacks



Services



Containers



Images



Networks



Volumes



Configs



Secrets



Swarm



SETTINGS

Extensions



Users



Endpoints



Registries



Settings



Portainer is connected to a node that is part of a Swarm cluster. Some resources located on other nodes in the cluster might not be available for management, have a look at our [agent setup](#) for more details.

### Endpoint info

Endpoint primary 2 1.5 GB - Swarm 18.03.1-ce

URL /var/run/docker.sock

Tags -

[Go to cluster visualizer](#)



0  
Stacks



1  
Service




2  
Containers

1 running  
1 stopped



5  
Images

315.3 MB

  
Home  
PRIMARY  
Dashboard  
App Templates  
Stacks  
Services  
Containers  
Images  
Networks  
Volumes  
Configs  
Secrets  
Swarm  
SETTINGSContainer list 










Containers

 Portainer support  admin[my account](#) [log out](#)

Containers Columns Settings

[Start](#) [Stop](#) [Kill](#) [Restart](#) [Pause](#) [Resume](#) [Remove](#) [+ Add container](#)

Search...

<input type="checkbox"/>	Name	State  Filter 	Quick actions	Stack	Image	Created	IP Address	Published Ports	Ownership
<input type="checkbox"/>	portainer.1.lno5rlkxzob0u0sp...	running	   	-	portainer/portainer:latest	2019-06-07 11:37:31	10.255.0.6	-	 administrators
<input type="checkbox"/>	portainer.1.qqbbvv2z8xefc4ws0...	stopped		-	portainer/portainer:latest	2019-06-07 11:32:26	-	-	 administrators

Items per page 10

portainer.io

- Home
- PRIMARY
- Dashboard
- App Templates
- Stacks
- Services
- Containers**
- Images
- Networks
- Volumes
- Configs
- Secrets
- Swarm
- SETTINGS
- Extensions
- Users
- Endpoints
- Registries

## Container details

Containers > portainer.1.qqbbvv2z8xefc4ws0i5ywqgl3

Portainer support admin

[my account](#) [log out](#)

### Actions

[Start](#) [Stop](#) [Kill](#) [Restart](#) [Pause](#) [Resume](#) [Remove](#)

### Container status

ID	3ef81ba7b0a74d778c8babd51c408de3b71cebd750bdbd070d897326bad9d1ef
Name	portainer.1.qqbbvv2z8xefc4ws0i5ywqgl3 <a href="#">✎</a>
Status	❤ Stopped for 10 minutes with exit code 1
Created	2019-06-07 11:32:26
Finished	2019-06-07 11:37:27

[Logs](#) [Inspect](#) [Stats](#) [\\_ Console](#) [Attach](#)

Image `nginx@sha256:d004263c915b985384f950520de619425126b977ec7bfcf5d5222d97f2c26be2`

Port configuration `0.0.0.0:443 → 443/tcp`  
`0.0.0.0:80 → 80/tcp`

CMD `nginx -g daemon off;`

ENV	Value
PATH	<code>/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin</code>
NGINX_VERSION	<code>1.17.2</code>
NJS_VERSION	<code>0.3.3</code>
PKG_RELEASE	<code>1~buster</code>

Labels	Value
<code>com.docker.compose.config-hash</code>	<code>9d898bd4e54f835b632e2d87bc20dae943614df78caafb5dd9c70e8a4b303854</code>
<code>com.docker.compose.container-number</code>	<code>1</code>
<code>com.docker.compose.oneoff</code>	<code>False</code>
<code>com.docker.compose.project</code>	<code>blog</code>
<code>com.docker.compose.service</code>	<code>nginx</code>
<code>com.docker.compose.version</code>	<code>1.24.1</code>
<code>maintainer</code>	<code>NGINX Docker Maintainers &lt;docker-maint@nginx.com&gt;</code>

## Volumes

Settings

Remove

+ Add volume

Search...

 Name ↓  
Filter ▼

Stack

Driver

Mount point

Created

Ownership

<input type="checkbox"/> <a href="#">blog_ghost-volume</a>	blog	local	/var/lib/docker/volumes/blog_ghost-volume/_data	2019-08-16 13:57:03	 administrators
<input type="checkbox"/> <a href="#">blog_mysql-volume</a>	blog	local	/var/lib/docker/volumes/blog_mysql-volume/_data	2019-08-16 13:57:09	 administrators
<input type="checkbox"/> <a href="#">blog_nginx-volume</a>	blog	local	/var/lib/docker/volumes/blog_nginx-volume/_data	2019-08-16 13:53:11	 administrators
<input type="checkbox"/> <a href="#">blog_portainer_data</a>	blog	local	/var/lib/docker/volumes/blog_portainer_data/_data	2019-08-17 03:19:07	 administrators



## Networks

[Settings](#)[Remove](#)[+ Add network](#)

Search...

<input type="checkbox"/> Name ↓↑	Stack	Scope	Driver	Attachable	Internal	IPAM Driver	IPAM Subnet	IPAM Gateway	Ownership
<input type="checkbox"/> blog_default	blog	local	bridge	true	false	default	172.25.0.0/16	172.25.0.1	administrators
<input type="checkbox"/> blog_ghost	blog	local	bridge	true	false	default	172.22.0.0/16	172.22.0.1	administrators
<input type="checkbox"/> blog_mysql	blog	local	bridge	true	false	default	172.24.0.0/16	172.24.0.1	administrators
<input type="checkbox"/> blog_nginx	blog	local	bridge	true	false	default	172.23.0.0/16	172.23.0.1	administrators
<input type="checkbox"/> bridge	-	local	bridge	false	false	default	172.17.0.0/16	172.17.0.1	administrators
<input type="checkbox"/> busybox_default	busybox	local	bridge	true	false	default	172.21.0.0/16	172.21.0.1	administrators
<input type="checkbox"/> host	-	local	host	false	false	default	-	-	administrators
<input type="checkbox"/> none	-	local	null	false	false	default	-	-	administrators

## Image list

Images

 Pull image

Image

Registry

DockerHub

 Image name is required.

Note: if you don't specify the tag in the image name, `latest` will be used.

[Pull the image](#)

 Images

 Remove

[+ Build a new image](#)

[Import](#)

[Export](#)

Id  
Filter

Tags

Size

Created

sha256:11cd0b38bc3ceb958ffb2f9bd70be3...

Unused

alpine:latest

4.4 MB

2018-07-06 16:14:06

sha256:da27590081471f174bc3f8c5f74b63...

portainer/portainer:latest

75.4 MB

2019-06-04 06:22:27

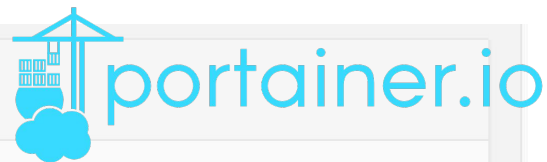
sha256:4e8db158f18dc71307f95260e532df...




Unused

redis:latest

83.4 MB

2018-08-04 04:40:22

 Image layers

Order ▾	Size	Layer
1	41.5 MB	ADD file:ce72656b2ad1b7552ba556c9a12aecdff82325d9938c28f7a3be6dd8bca7b5b1 in /
2	0 B	CMD ["bash"]
3	325.3 kB	RUN groupadd --gid 1000 node && useradd --uid 1000 --gid node --shell /bin/bash --create-home node
4	0 B	ENV NODE_VERSION=10.16.2
5	75.6 MB	RUN buildDeps='xz-utils' && ARCH= && dpkgArch="\$(dpkg --print-architecture)" && case "\${dpkgArch##*-}" in amd64) ... 
6	0 B	ENV YARN_VERSION=1.17.3
7	5.5 MB	RUN set -ex && for key in 6A010C5166006599AA17F08146C2130DFD2497F5 ; do gpg --batch --keyserver hkp://p80.pool.sks-... 
8	116 B	COPY file:238737301d47304174e4d24f4def935b29b3069c03c72ae8de97d94624382fce in /usr/local/bin/
9	0 B	ENTRYPOINT ["docker-entrypoint.sh"]
10	0 B	CMD ["node"]
11	0 B	ENV GOSU_VERSION=1.10
12	1.2 MB	RUN set -x && wget -O /usr/local/bin/gosu "https://github.com/tianon/gosu/releases/download/\$GOSU_VERSION /gosu-\$(dpkg --print-... 

## Container statistics

Containers > nginx > Stats

 Portainer support  admin

[my account](#) [log out](#)

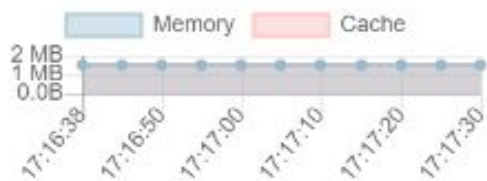
### About statistics

This view displays real-time statistics about the container `nginx` as well as a list of the running processes inside this container.

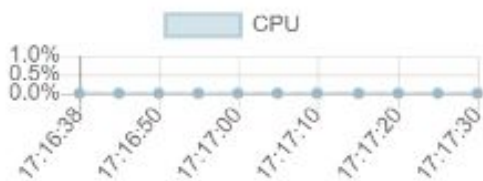
Refresh rate

5s

### Memory usage



### CPU usage



### Network usage (aggregate)



### Processes

# <https://www.katacoda.com/portainer/scenarios/deploying-to-swarm>

## Deploying Portainer to Docker Swarm Cluster

◀ Step 1 of 5 ▶

### Step 1 - Deploy Swarm Cluster

This scenario deploys Portainer to a Swarm cluster. A two-node cluster has been created for you. If you would like more information on how this was created, complete the Katacoda scenario on [Getting Started With Swarm Mode](#).

You can view the status of the Swarm cluster with the command `docker node ls`

CONTINUE

Terminal

Portainer UI



Katacoda

### Connecting to Port 9000

We're currently trying to connect to a HTTP service running on 9000. Services can sometimes take a few moments to start, even upto five minutes.

### Display a different port

If the service is running on a different port then please enter it below

Display Port



# Conclusions

- **Challenges** in containers and architectures distributed.
- **Metrics** that we can use to measure container performance.
- Tools for monitoring and management of containers such as **cadvisor**, **sysdig** and **portainer**.
- **Rancher** as a platform for the administration of Kubernetes.





Community Experience Distilled

# Monitoring Docker

Monitor your Docker containers and their apps using various native and third-party tools with the help of this exclusive guide!

Russ McKendrick

**[PACKT]** open source\*  
PUBLISHING community experience distilled

## Monitoring Docker

This book will show you how monitoring containers and keeping a keen eye on the working of applications helps improve the overall performance of the applications that run on Docker.

The book covers monitoring containers using Docker's native monitoring functions, various plugins, as well as third-party tools that help in monitoring. We'll start with how to obtain detailed statistics for active containers, resources consumed, and container behavior. We also show you how to use these stats to improve the overall performance of the system. Next, you will learn how to use Sysdig to both view your containers' performance metrics in real time and record sessions to query later. By the end of this book, you will have a complete knowledge of how to implement monitoring for your containerized applications and make the most of the metrics you are collecting.

### Who this book is written for

This book is for DevOps engineers and system administrators who manage Docker containers and want to better manage these containers using expert techniques and methods, and also improve the way they maintain their applications built on Docker.



### What you will learn from this book

- Discover the tools built into Docker to gain an insight into your container's performance
- Augment Docker's built-in tools with modern tools such as cAdvisor from Google, Sysdig by Draios, and Soundcloud's Prometheus
- Integrate the monitoring of your containers with more traditional monitoring solutions, such as Zabbix
- Take advantage of the various SaaS offerings from third parties to move monitoring away from your local infrastructure and into the cloud
- Discover the various ways to ship your application's logs from the container to a central logging service
- Get the most out of your application and resources with the right implementation of your monitoring method

\$ 39.99 US  
£ 25.99 UK



Prices do not include local sales tax or VAT where applicable

**[PACKT]** open source\*  
PUBLISHING community experience distilled

Visit [www.PacktPub.com](http://www.PacktPub.com) for books, eBooks, code, downloads, and PacktLib.





Finish

Thank You

