JavaCro

Unikernelize your Java Application

Mario Zagar

mario.zagar@infobip.com

infobip

JavaCro

seamless interaction of businesses and people through mobile

Java, .NET, nodejs, reactjs

microservices

continuos delivery practitioners

25+ agile dev teams

100+ deploys daily / 6 datacenters

300M+ messages per day

JavaCro

Agenda

what are unikernels
why should you care
how does java fit in
build java unikernel app

What are unikernels?

“specialised, single-address-space machine images constructed by using library operating systems.”

App code

+

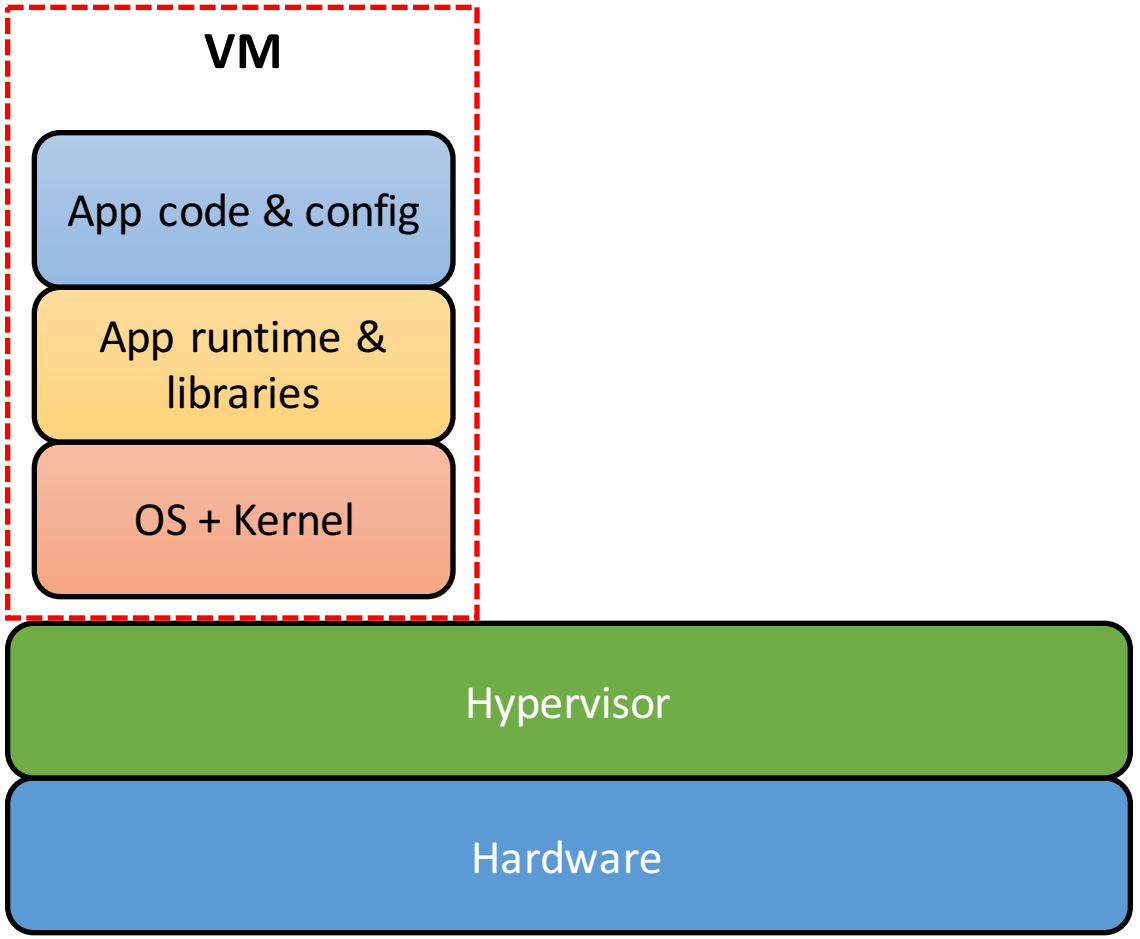
OS parts ***actually*** needed

=

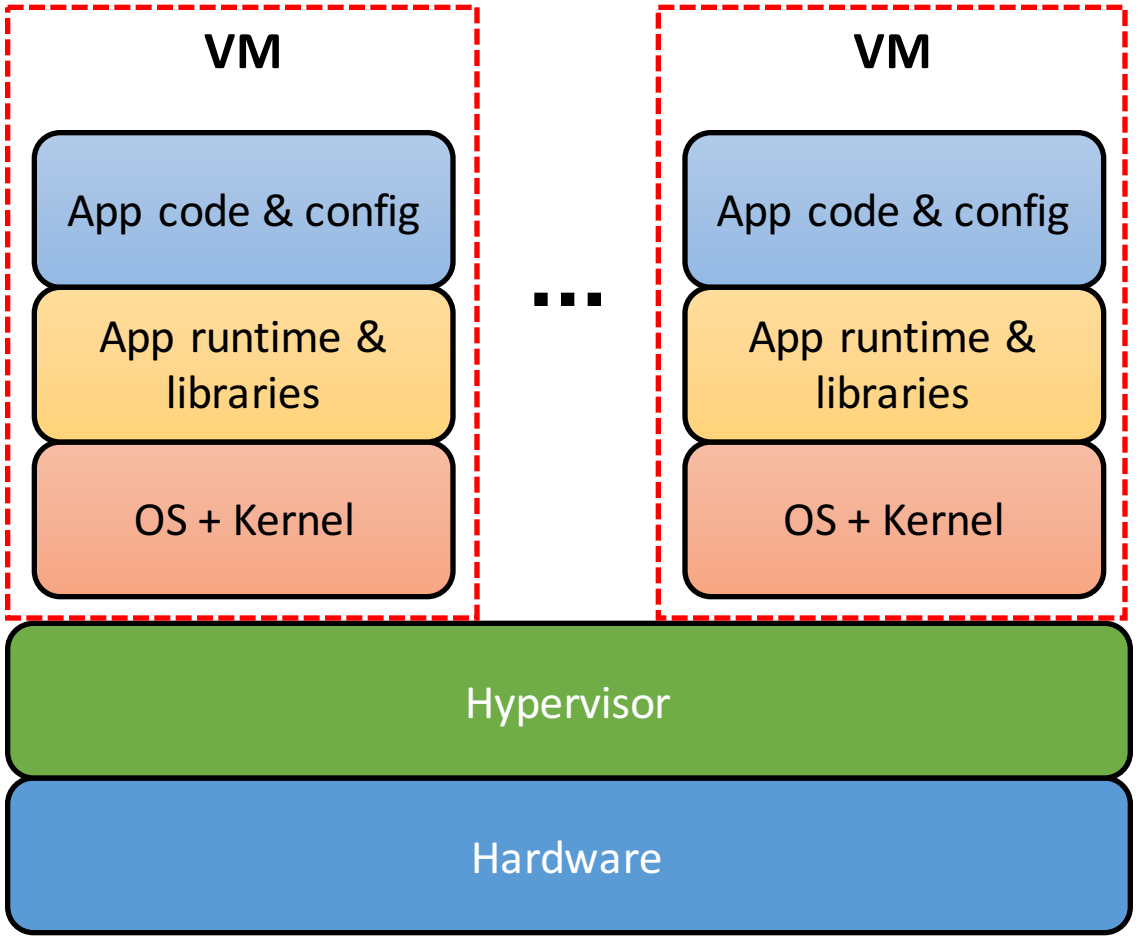
unikernel

Typical cloud deployment

Typical cloud deployment

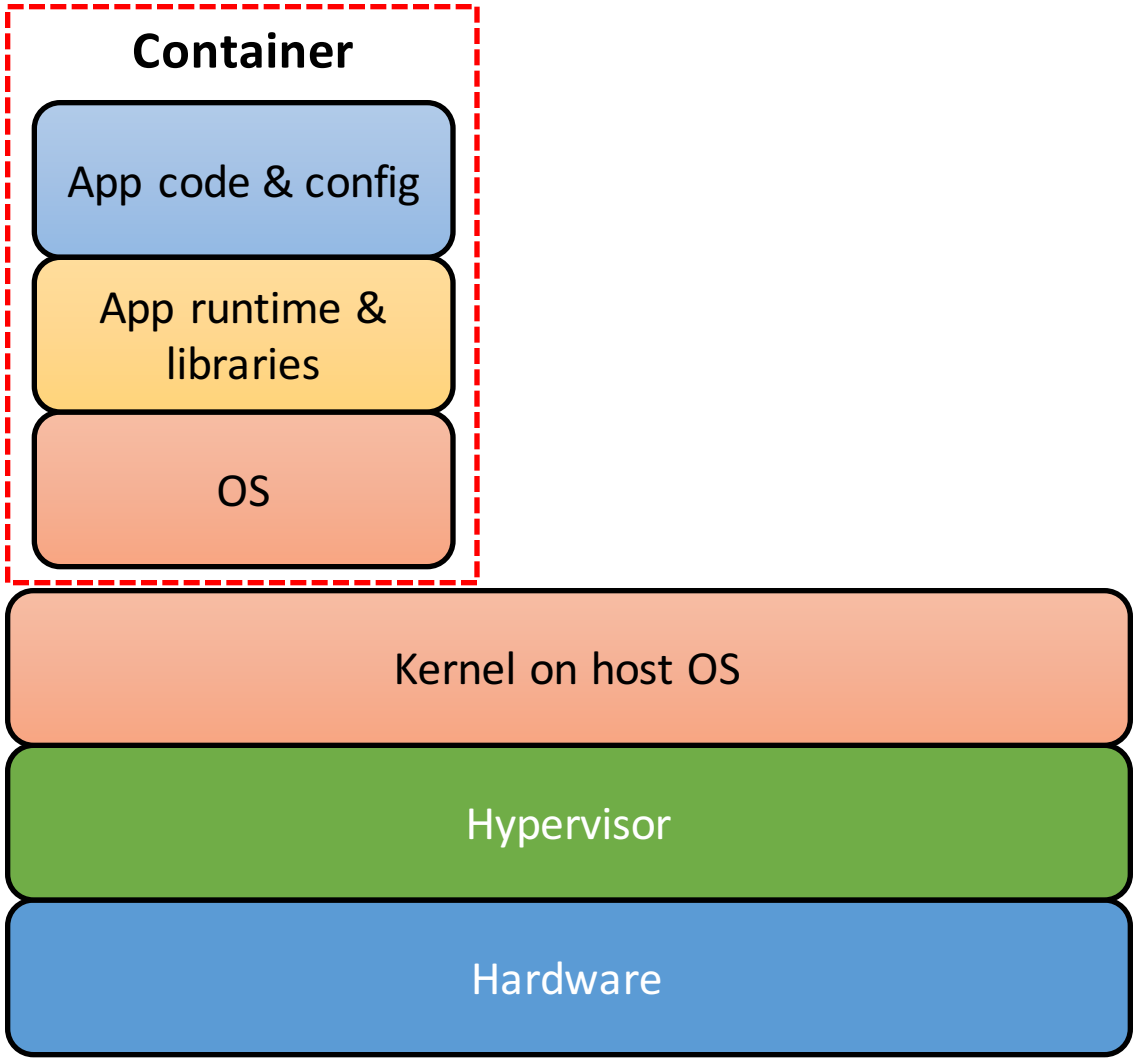


Typical cloud deployment

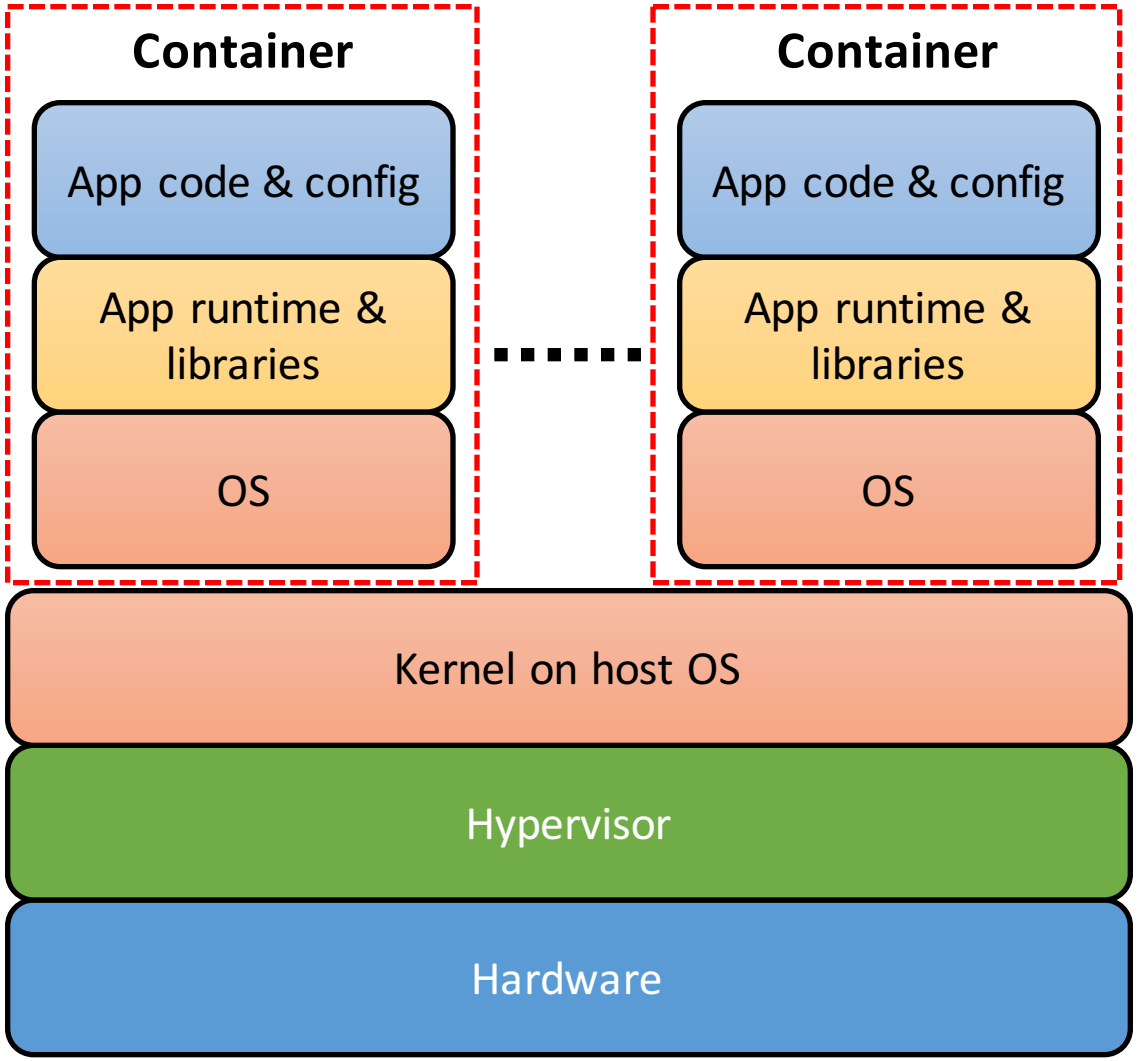


Containerized deployment

Containerized deployment



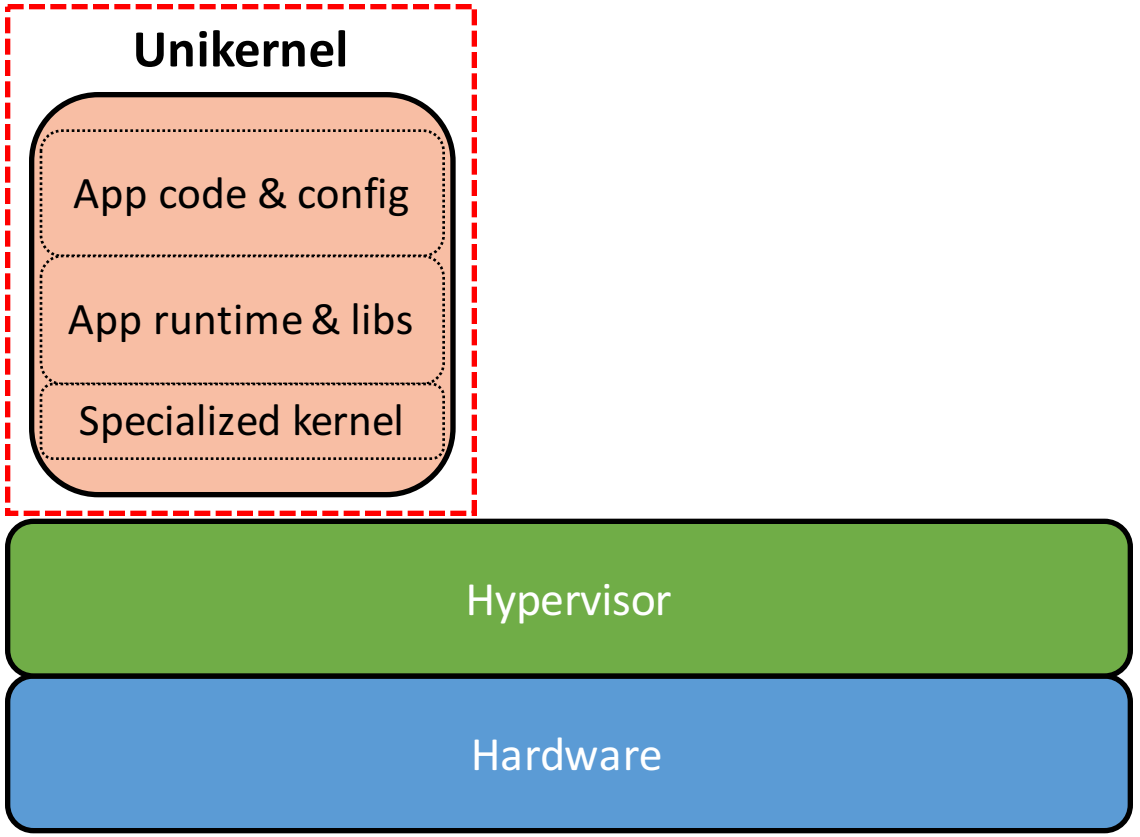
Containerized deployment



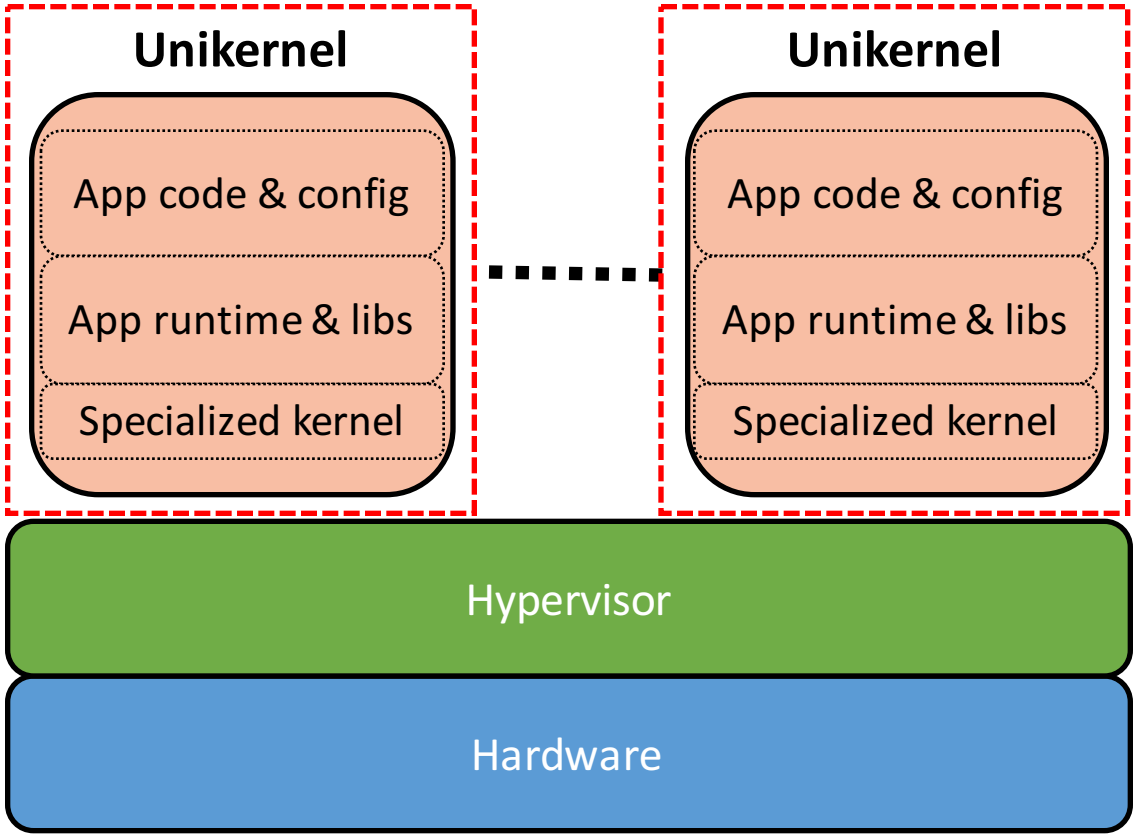
JavaCro

What if…

Unikernel deployment

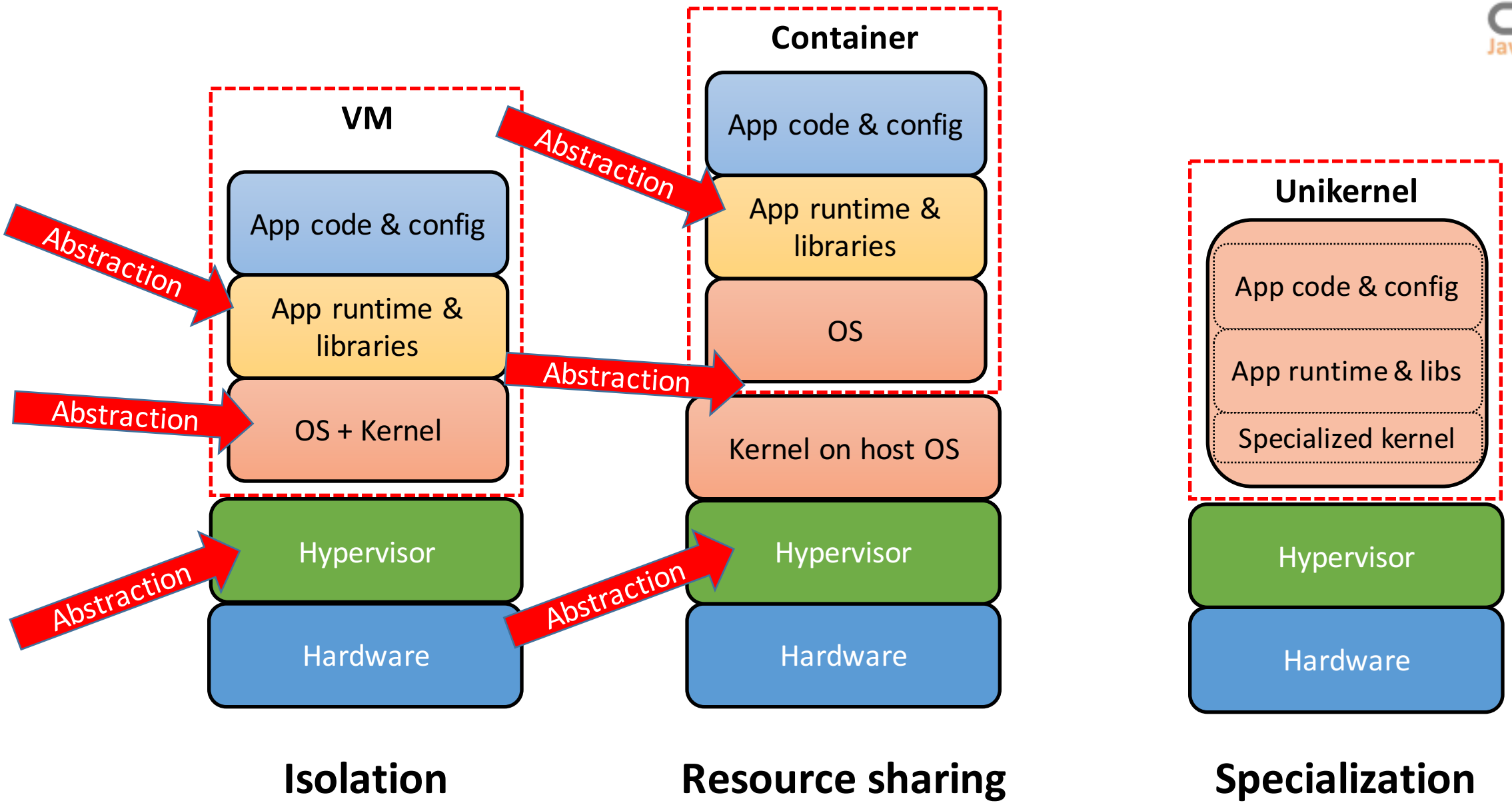


Unikernel deployment



JavaCro

immutable server


VM
App code & config
App runtime & libraries
OS + Kernel
Hypervisor
Hardware
Container
App code & config
App runtime & libraries
OS
Kernel on host OS
Hypervisor
Hardware
Unikernel
App code & config
App runtime & libs
Specialized kernel
Hypervisor
Hardware
Abstraction
Abstraction
Abstraction
Abstraction
Abstraction
Abstraction
Isolation
Resource sharing
Specialization

"just enough" of the OS to run your code, nothing more.

JavaCro

Why should I use this?

JavaCro

Why should I use this?

Small footprint

JavaCro

Why should I use this?

Small footprint

Fast boot

JavaCro

Why should I use this?

Small footprint

Fast boot

Highly optimized

JavaCro

Why should I use this?

Small footprint

Fast boot

Highly optimized

Reduced attack surface

No, really… why should I use this?

same amount of work
on less powerful VMs…
…**saves $$$**

JavaCro

Cons ?

JavaCro

Cons ?

no fork() or exec()

JavaCro

Cons ?

no fork() or exec()

hard to debug

JavaCro

Cons ?

no fork() or exec()

hard to debug

“secure” is just a matter of time

JavaCro

Cons ?

no fork() or exec()

hard to debug

“secure” is just a matter of time

troubleshooting tools

JavaCro

What’s out there ?

MIRAGE OS

IncludeOS

#include <os> // Literally

JavaCro

HaLVM

runtime.js

JAVASCRIPT LIBRARY OPERATING SYSTEM FOR THE CLOUD

Erlang on Xen

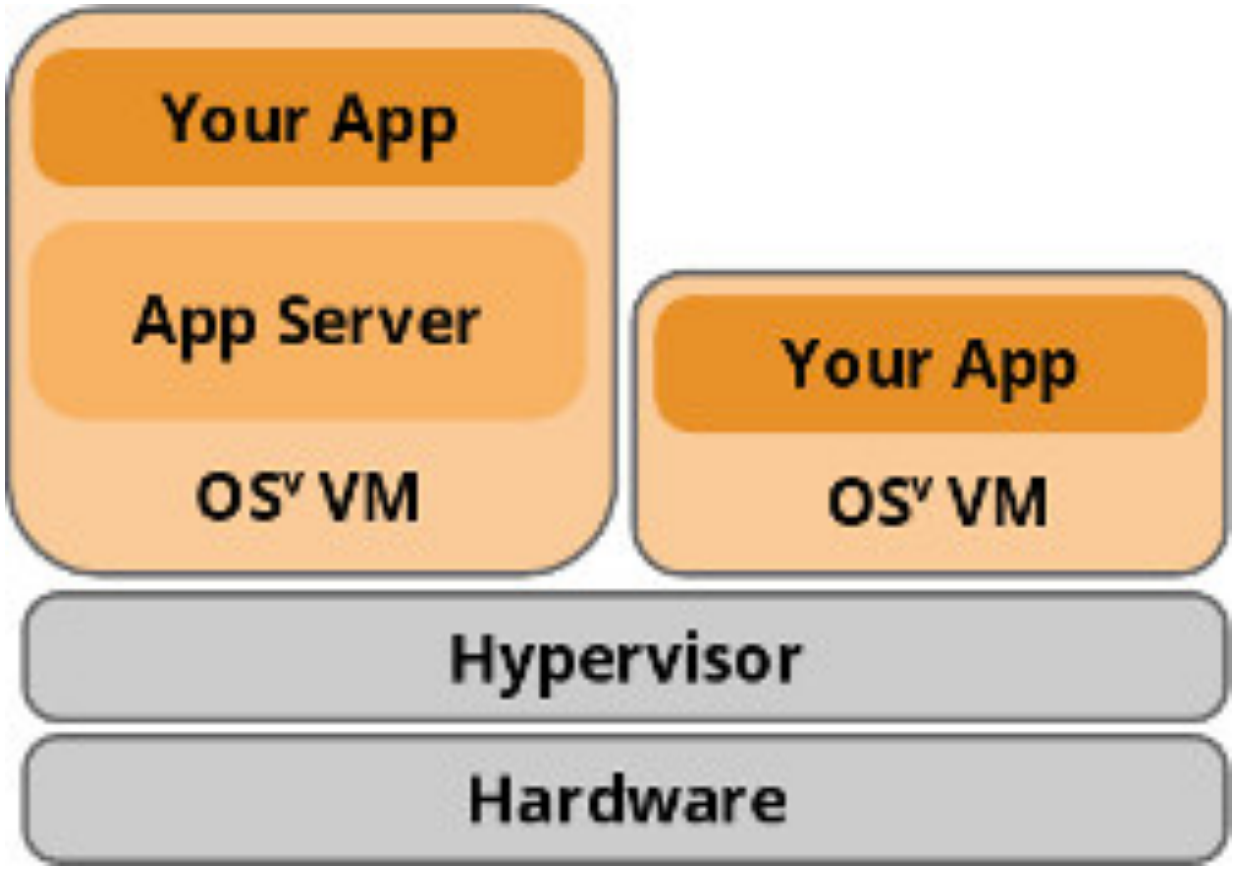
- at the heart of super-elastic clouds

JavaCro

How does java fit in ?


OSv.
designed for the cloud
Beta

OSv
designed for the cloud
Beta



new OS designed for the cloud

C++

native Linux apps, JVM

single address space

boots < 1 sec

Lower administration & OS support costs

JavaCro

Hypervisor support

KVM, XEN - fully

VMW, VBox – exp

JavaCro

Cloud support

EC2 - fully

GCE – exp

a little “fatter”
min image size ~20MB
Java image ~120MB

JavaCro

REST API
JMX/Jolokia API
cloud-init

Benchmarks

redis

50% more TPS

memcached

20% more req/sec

TCP

20% better latency

How to build a java unikernel app?

JavaCro

Capstan build tool
(feels dockerish)

JavaCro

Capstanfile

Capstanfile

```
base: cloudius/osv-openjdk

build: mvn package

files: /app.jar: target/app.jar

cmdline: /java.so -jar /app.jar
```

JavaCro

$ capstan build -p vbox

JavaCro

```
$ mvn package
$ capstan run target/app.jar
```


STOP!
DEMO TIME!

JavaCro

So where is this all going ?

few users

hard to ship

hard to build

hard to run

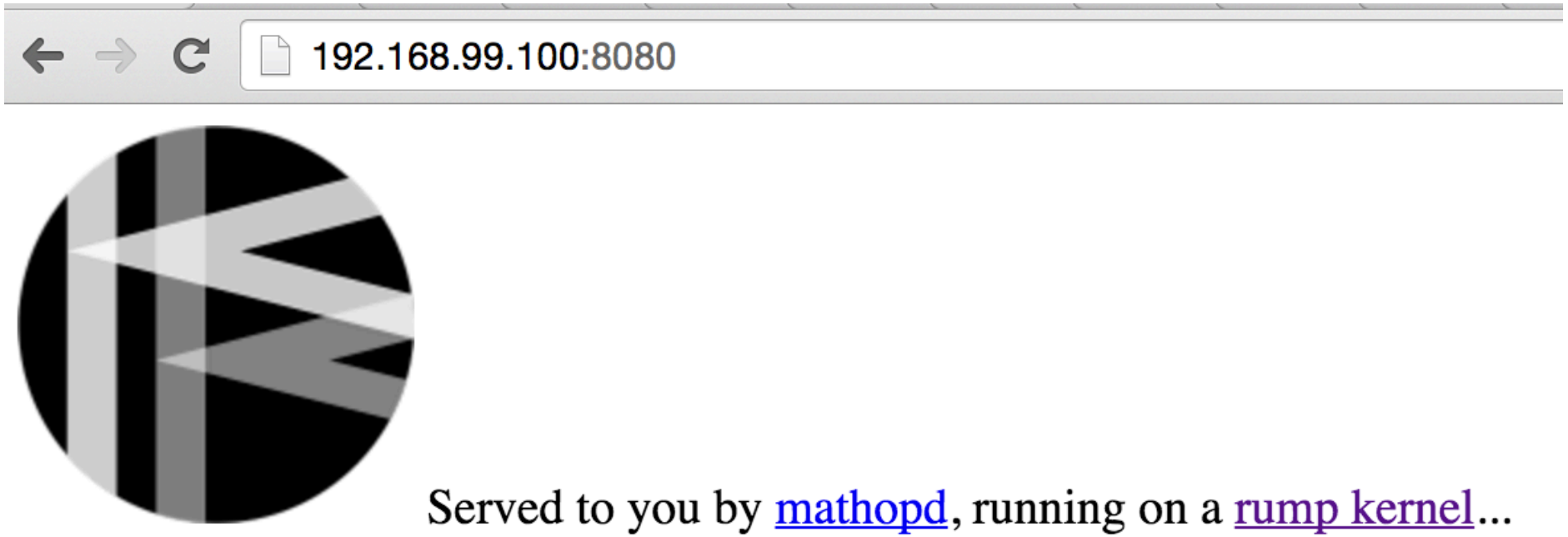
Community


docker
Buys
Unikernel Systems

JavaCro

unikernels + docker

```
mzagar@marios-mbp ~/work/osv/groovy-example $ docker run --rm -ti \
>       --device /dev/net/tun:/dev/net/tun \
>       --cap-add NET_ADMIN \
>       -p 8080:80 \
>       mato/unikernel-mathopd
```


192.168.99.100:8080
Served to you by mathopd, running on a rump kernel...

”next big thing in the cloud”

JavaCro

Thank you!

JavaCro

Building java osv app for virtualbox – demo slides

```
mzagar@marios-mbp ~/work/osv/groovy-example $ ls -la
total 16
drwxr-xr-x  4 mzagar  staff  136 Apr 22 22:51 .
drwxr-xr-x  8 mzagar  staff  272 Apr 21 23:07 ..
-rw-r--r--  1 mzagar  staff  637 Apr 19 08:03 Capstanfile
-rw-r--r--  1 mzagar  staff   95 Apr 19 08:00 app.groovy
```

```
mzagar@marios-mbp ~/work/osv/groovy-example $ cat app.groovy
@RestController
class App {
        @RequestMapping('/hello')
        String hello() {
                return "world"
        }
}
```

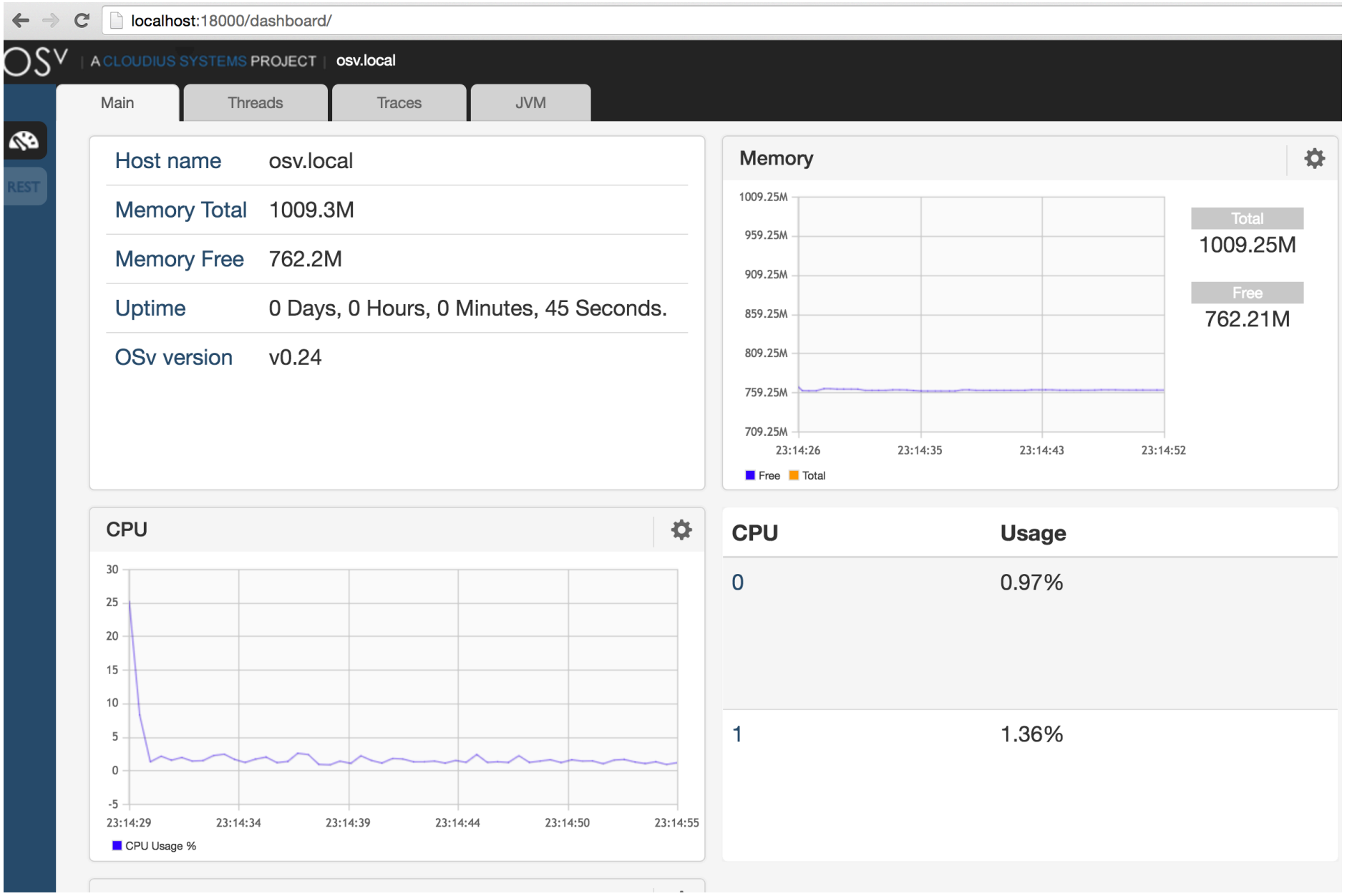
```
mzagar@marios-mbp ~/work/osv/groovy-example $ cat Capstanfile
base: cloudius/osv-openjdk

cmdline: /java.so -jar /app.jar

build: spring jar app.jar *

files:
  /app.jar: app.jar
```

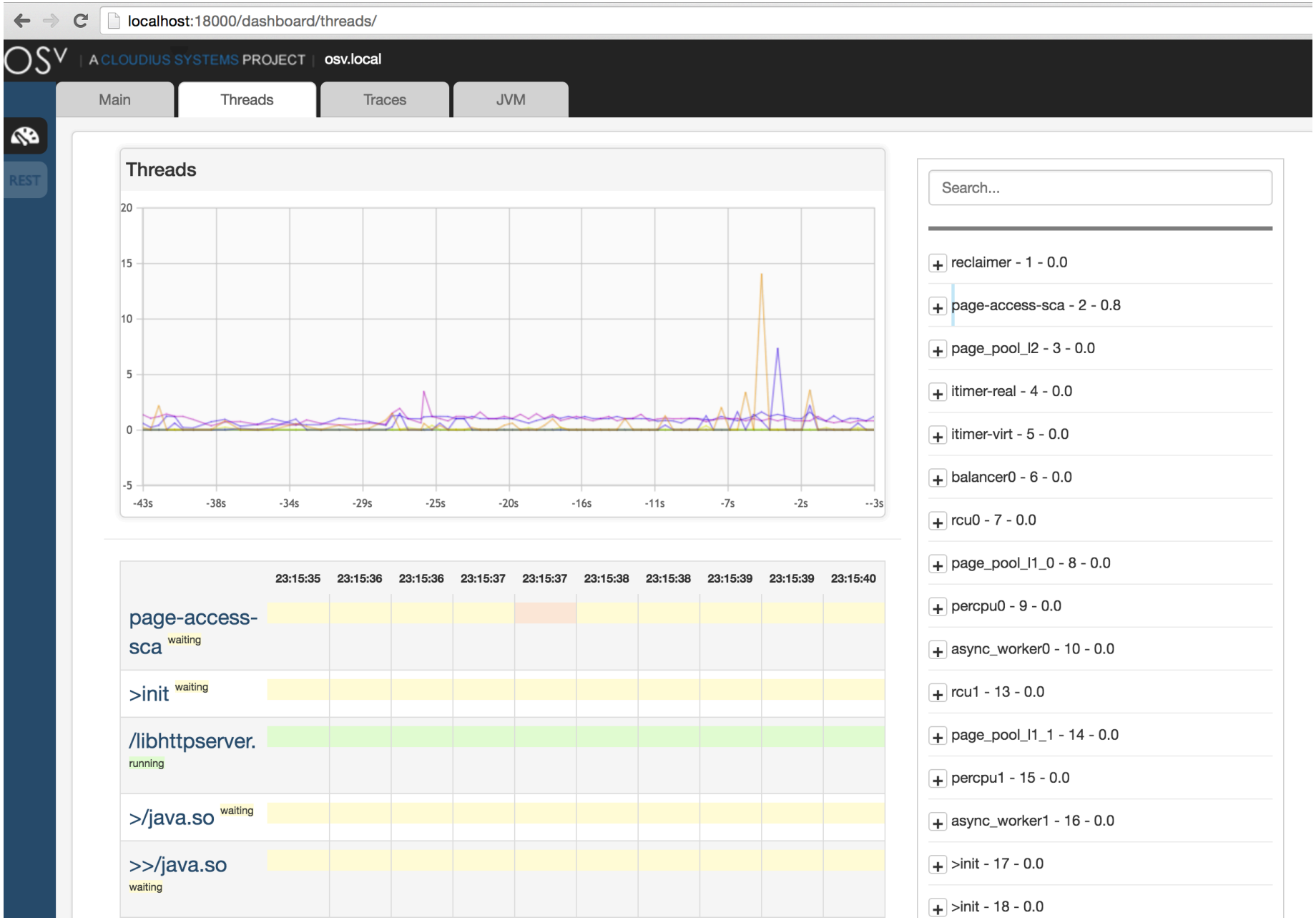
```
mzagar@marios-mbp ~/work/osv/groovy-example $ capstan build
Building groovy-example...
Uploading files...
 1 / 1 [==========================================================
```


```
mzagar@marios-mbp ~/work/osv/groovy-example $ capstan run -f "18080:8080"
Created instance: groovy-example
OSv v0.24
eth0: 10.0.2.15


  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v1.2.6.RELEASE)

2016-04-22 21:02:13.250  INFO 0 --- [       Thread-3] .b.c.j.PackagedSpringApp
ionLauncher on osv.local with PID 0 (/app.jar started by osv in /)
```

```
mzagar@marios-mbp ~ $ curl -v localhost:18080/hello
*   Trying ::1...
* connect to ::1 port 18080 failed: Connection refused
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 18080 (#0)
> GET /hello HTTP/1.1
> Host: localhost:18080
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: Apache-Coyote/1.1
< Content-Type: text/plain;charset=UTF-8
< Content-Length: 5
< Date: Fri, 22 Apr 2016 21:04:22 GMT
<
* Connection #0 to host localhost left intact
world%
```

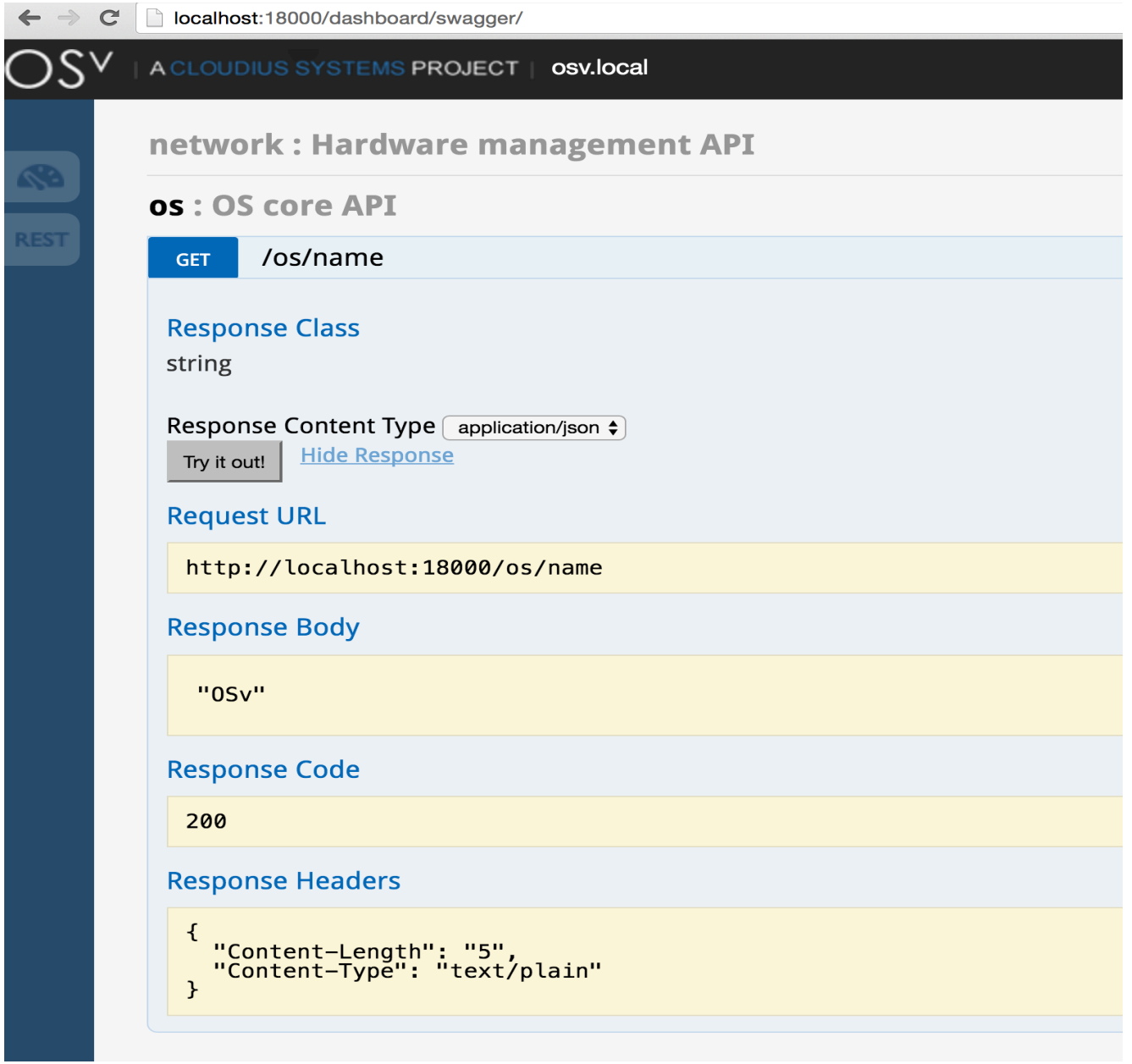
```
mzagar@marios-mbp ~ $ ab -n 10000 -c 10 127.0.0.1:18080/hello
Concurrency Level:      10
Time taken for tests:   5.273 seconds
Complete requests:      10000
Failed requests:        0
Total transferred:      1660000 bytes
HTML transferred:       50000 bytes
Requests per second:    1896.28 [#/sec] (mean)
Time per request:       5.273 [ms] (mean)
Time per request:       0.527 [ms] (mean, across all concurrent requests)
Transfer rate:          307.40 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.0      0       0
Processing:     1    5   2.4      5      34
Waiting:        1    5   2.3      5      33
Total:          1    5   2.4      5      34
```


localhost:18000/dashboard/
OSv
A CLOUDIUS SYSTEMS PROJECT
osv.local
Main
Threads
Traces
JVM
REST
Host name
osv.local
Memory Total
1009.3M
Memory Free
762.2M
Uptime
0 Days, 0 Hours, 0 Minutes, 45 Seconds.
OSv version
v0.24
Memory
1009.25M
959.25M
909.25M
859.25M
809.25M
759.25M
709.25M
23:14:26
23:14:35
23:14:43
23:14:52
Free
Total
Total
1009.25M
Free
762.21M
CPU
30
25
20
15
10
5
0
-5
23:14:29
23:14:34
23:14:39
23:14:44
23:14:50
23:14:55
CPU Usage %
CPU
Usage
0
0.97%
1
1.36%


localhost:18000/dashboard/threads/
OSV | A CLOUDIUS SYSTEMS PROJECT | osv.local
Main
Threads
Traces
JVM
REST
Threads
20
15
10
5
0
-5
-43s
-38s
-34s
-29s
-25s
-20s
-16s
-11s
-7s
-2s
--3s
23:15:35
23:15:36
23:15:36
23:15:37
23:15:37
23:15:38
23:15:38
23:15:39
23:15:39
23:15:40
page-access-sca waiting
>init waiting
/libhttpserver. running
>/java.so waiting
>>/java.so waiting
Search...
reclaimer - 1 - 0.0
page-access-sca - 2 - 0.8
page_pool_l2 - 3 - 0.0
itimer-real - 4 - 0.0
itimer-virt - 5 - 0.0
balancer0 - 6 - 0.0
rcu0 - 7 - 0.0
page_pool_l1_0 - 8 - 0.0
percpu0 - 9 - 0.0
async_worker0 - 10 - 0.0
rcu1 - 13 - 0.0
page_pool_l1_1 - 14 - 0.0
percpu1 - 15 - 0.0
async_worker1 - 16 - 0.0
>init - 17 - 0.0
>init - 18 - 0.0

JavaCro


```
{
    timestamp: 1461359837,
    status: 200,
  - request: {
        mbean: "java.lang:type=Memory",
        attribute: "HeapMemoryUsage",
        type: "read"
    },
  - value: {
        max: 518979584,
        committed: 49324032,
        init: 16769984,
        used: 30589192
    }
}
```


localhost:18000/dashboard/swagger/
OSv
A CLOUDIUS SYSTEMS PROJECT
osv.local
REST
network : Hardware management API
os : OS core API
GET
/os/name
Response Class
string
Response Content Type
application/json
Try it out!
Hide Response
Request URL
http://localhost:18000/os/name
Response Body
"OSv"
Response Code
200
Response Headers
{
"Content-Length": "5",
"Content-Type": "text/plain"
}