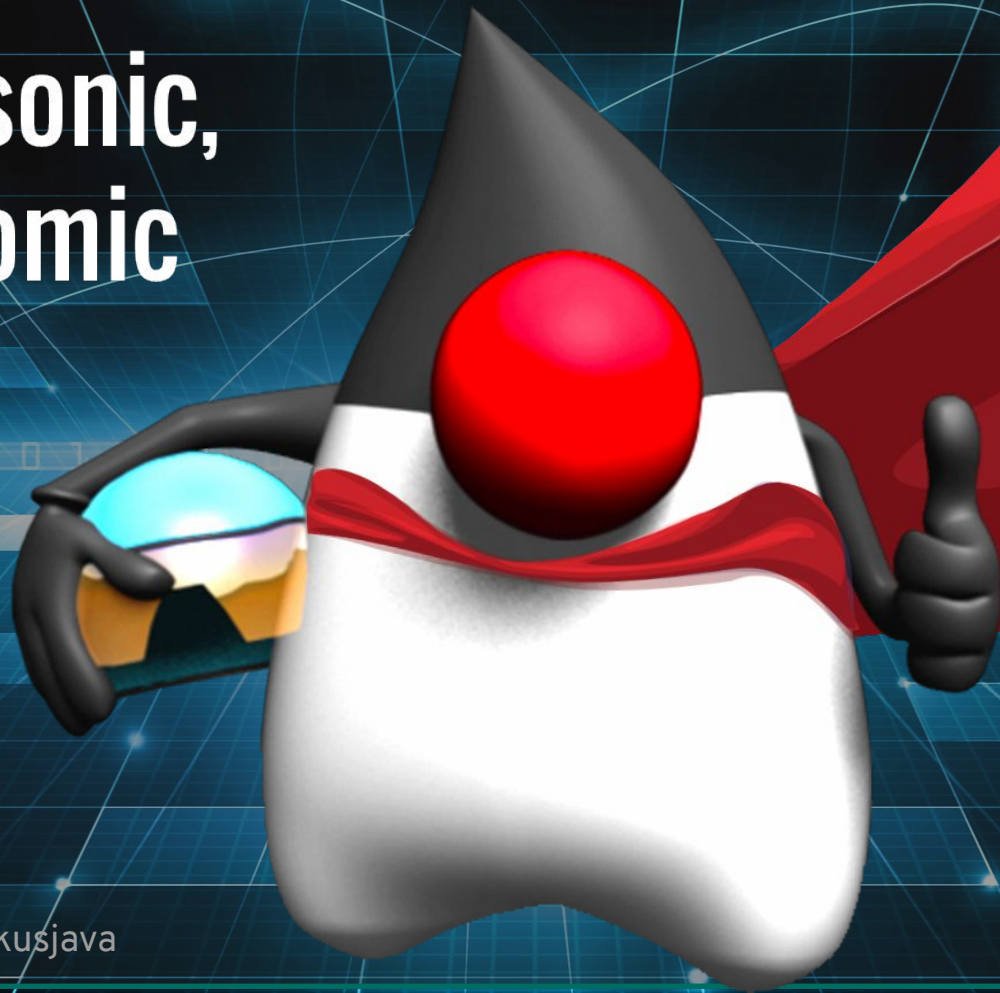


Supersonic, Subatomic Java

QUARKUS.IO

@bursutter bit.ly/quarkusjava



RED HAT
DEVELOPER

**“The reports of my
death are greatly
exaggerated.”**



@bursutter bit.ly/quarkusjava

"Today, Java is old hat."

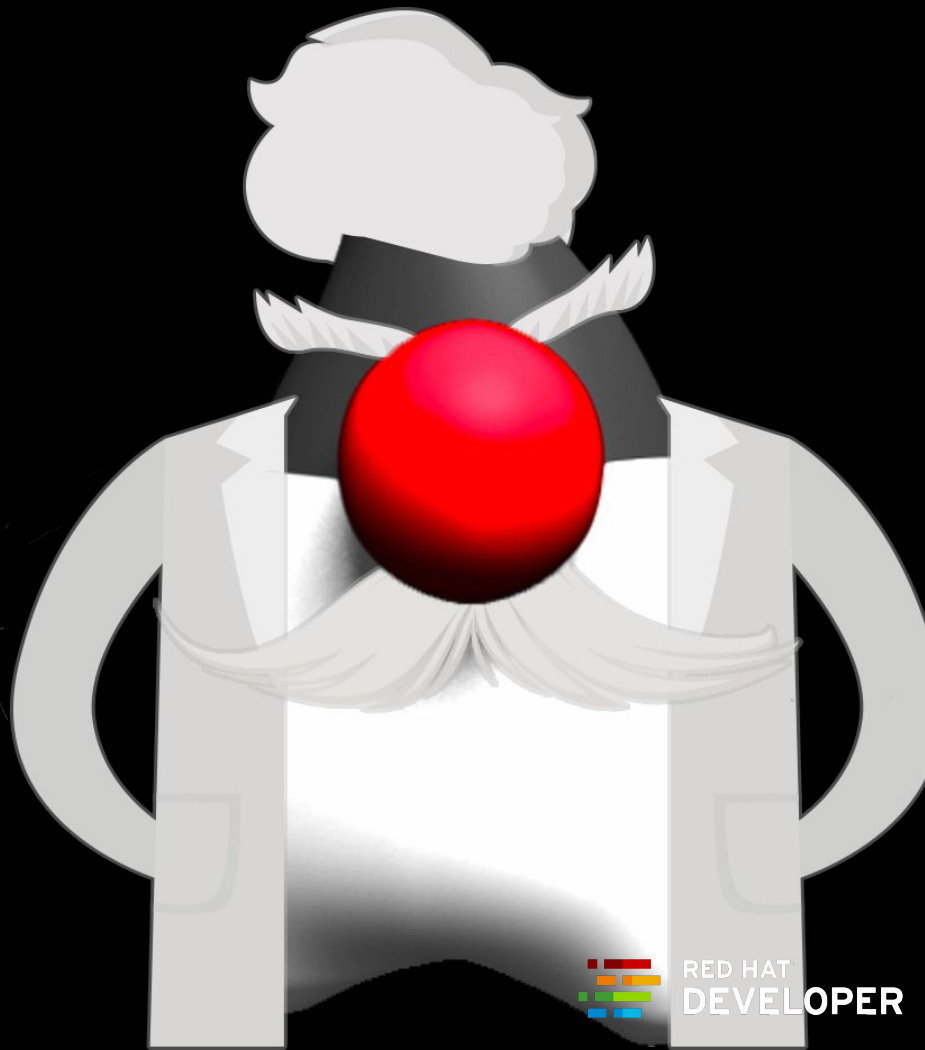
Computer World, May 22, 2006

FRANK HAYES ■ FRANKLY SPEAKING

Not Dead Yet

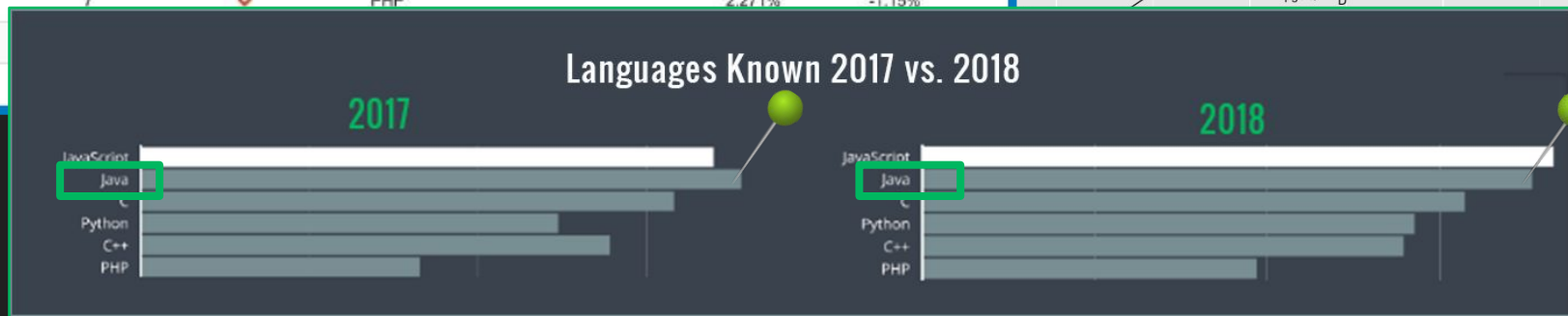
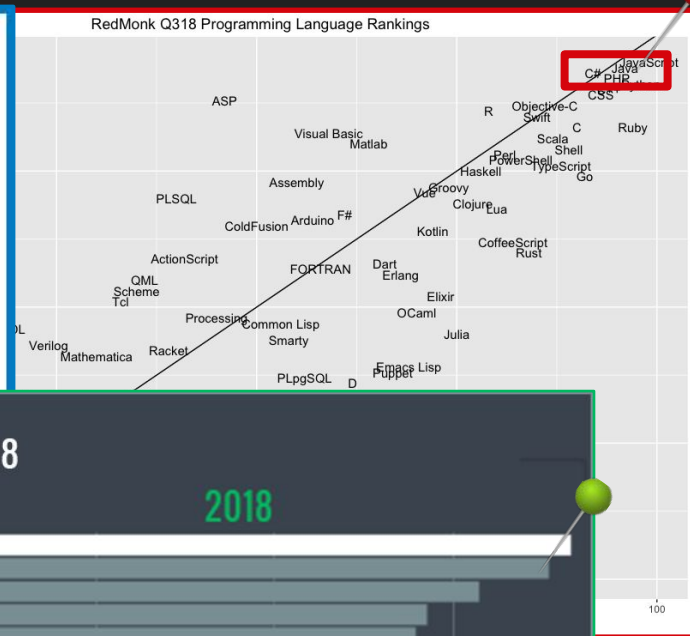
IS JAVA DEAD? Come on, seriously — why else would Sun Microsystems be offering it up to the open-source crowd? (See story, page 1.) A decade ago, Java was the hottest, most exciting thing in IT, a certified Windows-killer that was going to wipe out Microsoft's monopoly and revolutionize the way software was made, distributed and run. Today? Today, Java is old hat. It's been eclipsed by open-source, the *new* hottest thing in IT that's going to wipe out Microsoft's monopoly and revolutionize the way software is made, distributed and run.

@bursutter bit.ly/quarkusjava

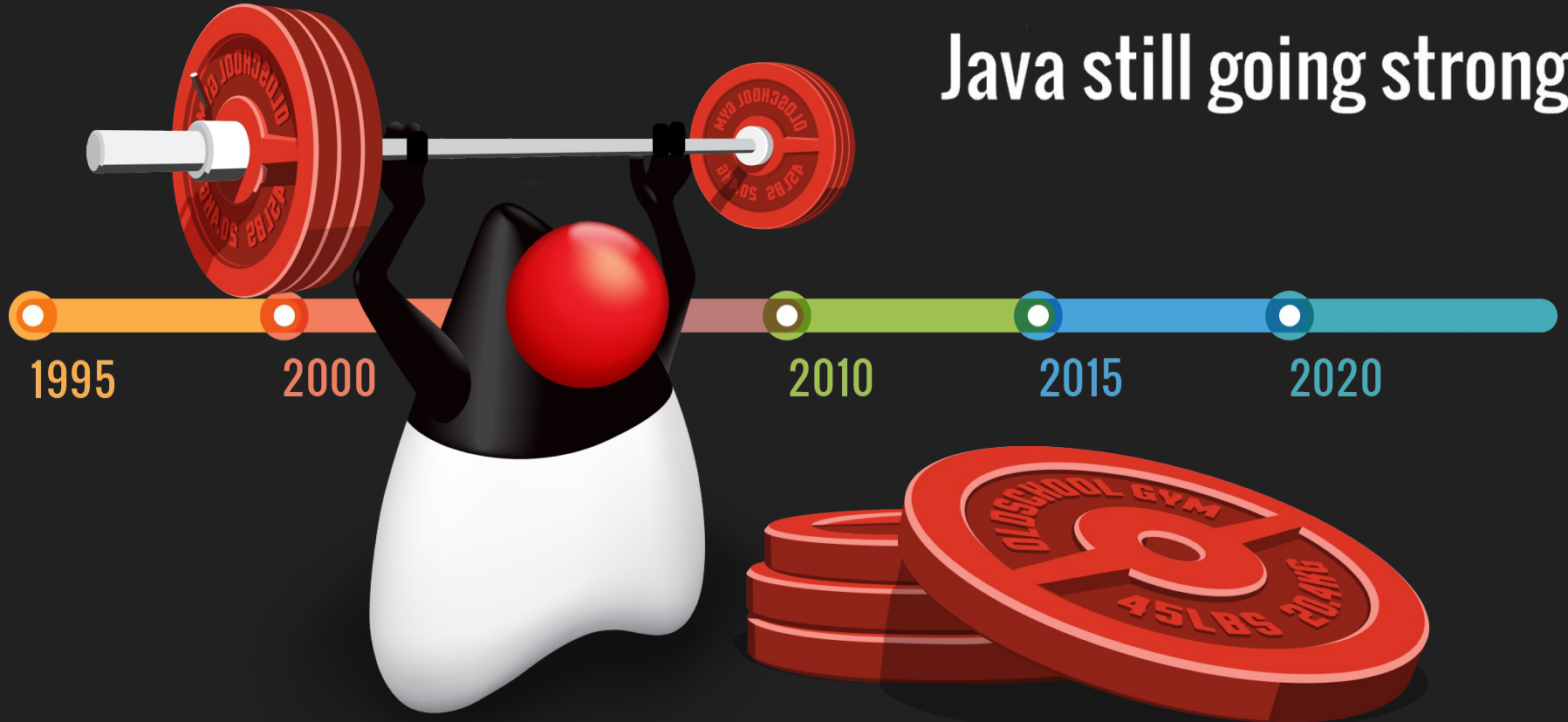


RED HAT
DEVELOPER

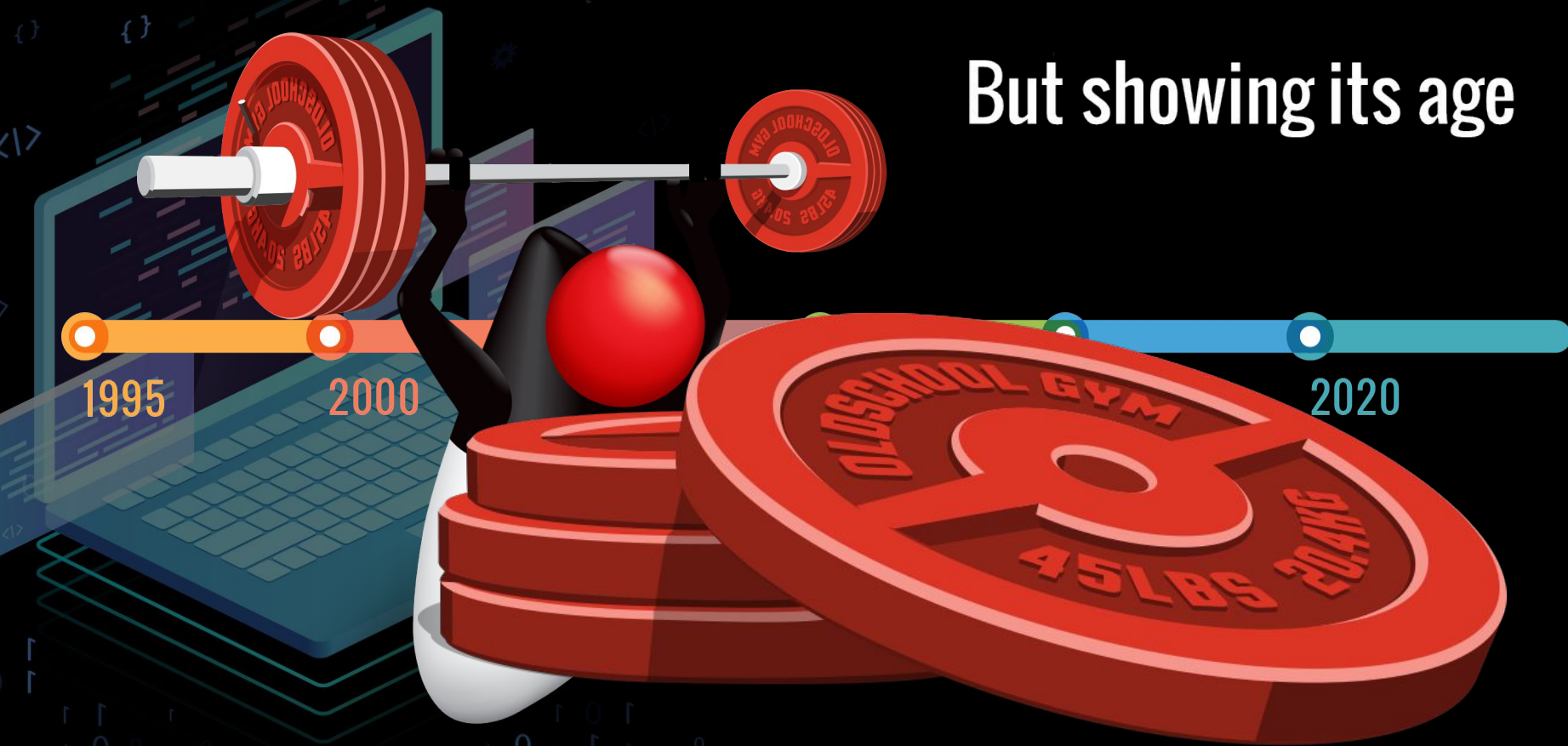
Feb 2019	Feb 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.876%	+0.89%
2	2		C	12.424%	+0.57%
3	4	▲	Python	7.574%	+2.41%
4	3	▼	C++	7.444%	+1.72%
5	6	▲	Visual Basic .NET	7.095%	+3.02%
6	8	▲	JavaScript	2.848%	-0.32%
7	5	▼	C#	2.346%	-1.61%
8	7	▼	PHP	2.271%	-1.15%
9					
10					



Java still going strong



But showing its age





Still, so much goodness ...



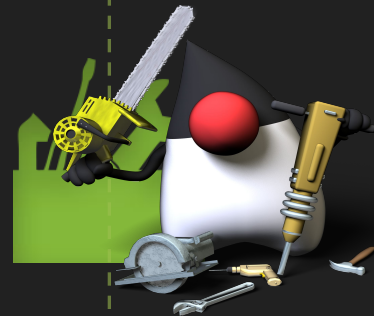


@burrsutter bit.ly/quarkusjava



RED HAT
DEVELOPER

But with so much to consume ...





In so many flavors ...



So so many flavors ...



Hey, is it getting a
little tight in here?



@burrsutter bit.ly/quarkusjava



@burrsutter bit.ly/quarkusjava



@burrsutter bit.ly/quarkusjava

1999

1999

USA Wins FIFA World Cup



1999



Cost of a Java-based Web App circa 1999

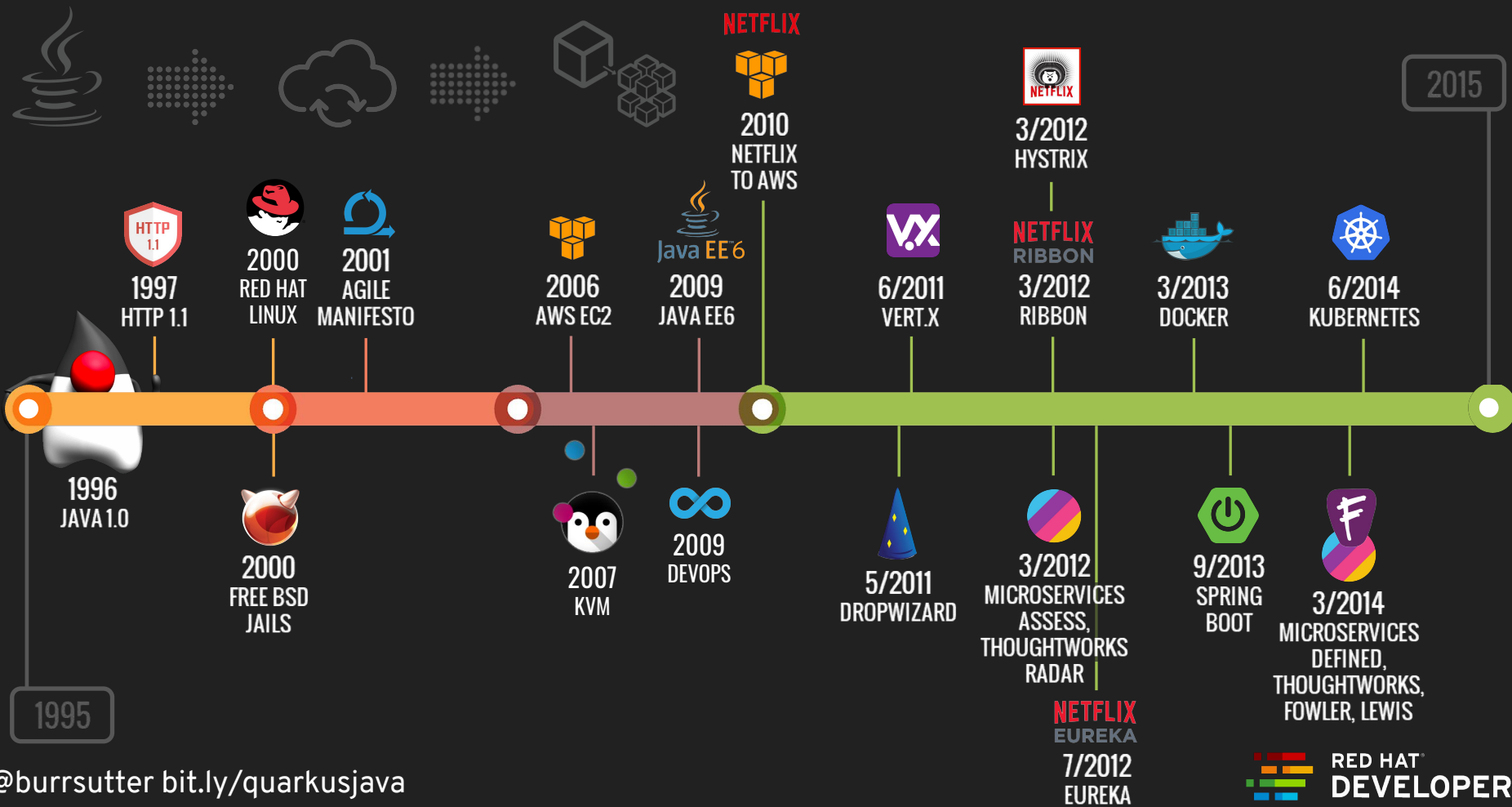
	\$18,000	Sun Sparc App Server Box (4 CPUs, 2GB of RAM)
+	\$60,000	BEA Weblogic
+	\$92,000	Sun Spark DB Server Box (8 CPUs)
+	\$243,000	Oracle RDBMS
+	\$50,000	Symantec Visual Café for 10 developers

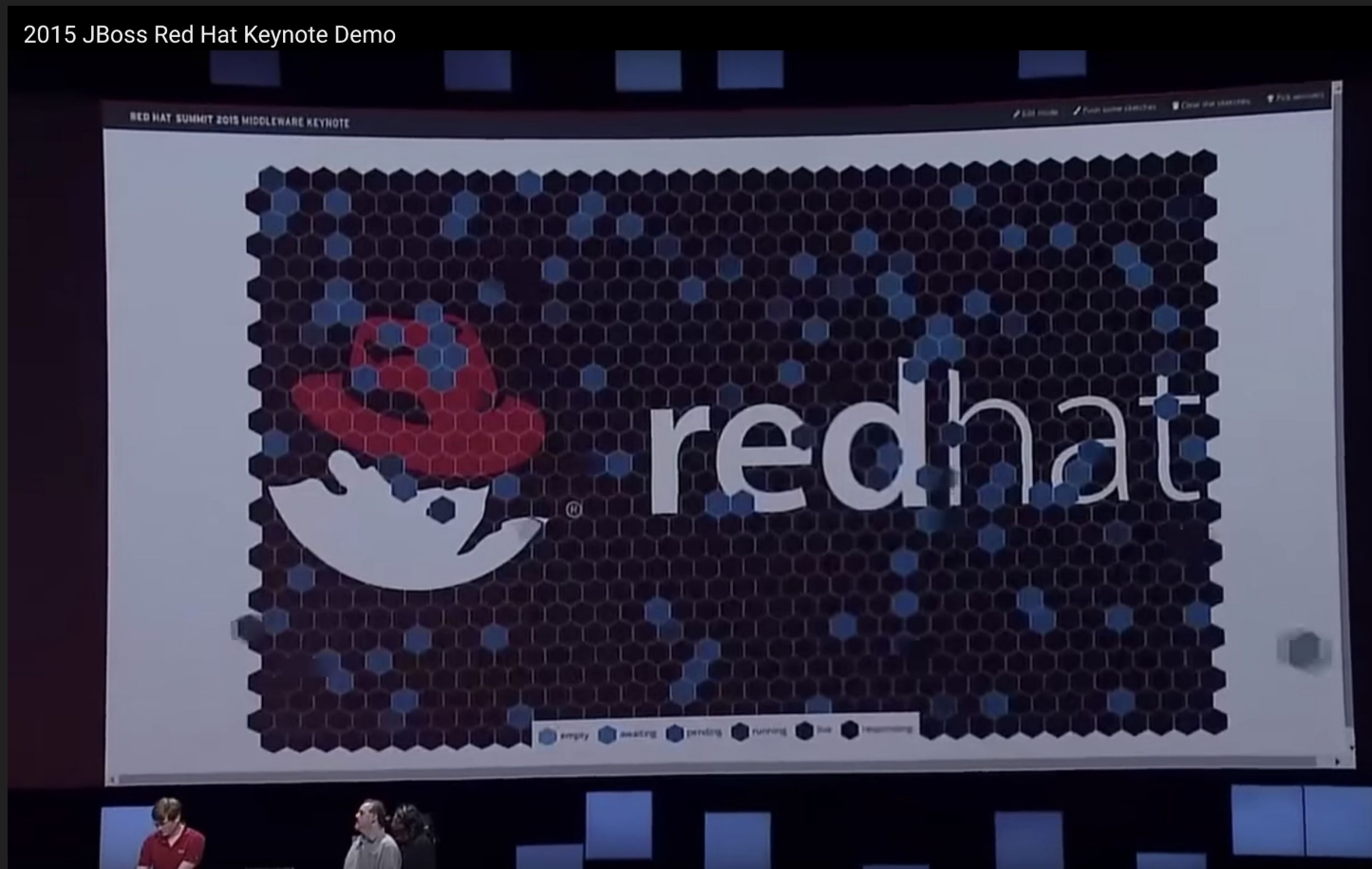
\$463,000 (capex) + ~\$80,000 annual maint (opex)

m5ad.4xlarge	16	N/A	64 GiB	2 x 300 NVMe SSD	\$0.824 per Hour
m5ad.12xlarge	48	N/A	192 GiB	2 x 900 NVMe SSD	\$2.472 per Hour
m5ad.24xlarge	96	N/A	384 GiB	4 x 900 NVMe SSD	\$4.944 per Hour
m5d.large	2	8	8 GiB	1 x 75 NVMe SSD	\$0.113 per Hour
m5d.xlarge	4	16	16 GiB	1 x 150 NVMe SSD	\$0.226 per Hour
m5d.2xlarge	8	31	32 GiB	1 x 300 NVMe SSD	\$0.452 per Hour

MEMORY	VCPUS	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	\$5/mo \$0.007/hr
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr
3 GB	1 vCPU	60 GB	3 TB	\$15/mo \$0.022/hr
2 GB	2 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
1 GB	3 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
4 GB	2 vCPUs	80 GB	4 TB	\$20/mo \$0.030/hr
8 GB	4 vCPUs	160 GB	5 TB	\$40/mo \$0.060/hr
16 GB	6 vCPUs	320 GB	6 TB	\$80/mo \$0.119/hr

INSTANCE	VCPU	RAM	TEMPORARY STORAGE	PAY AS YOU GO
D2 v3	2	8 GiB	50 GiB	\$0.096/hour
D4 v3	4	16 GiB	100 GiB	\$0.192/hour
D8 v3	8	32 GiB	200 GiB	\$0.384/hour





2015

Launch 1000+
Containers

Audience
Claims a
Container

<https://www.youtube.com/watch?v=GCtpncA0Ea0&feature=youtu.be&t=1031>

@burrsutter bit.ly/quarkusjava

“ “ IMHO you really should not be using Spring Boot for Lambdas ...

Spring is **designed for long running services**. Lambda's are literally functions that get called and then are gone. Different paradigm. Lambda's should be extremely tiny and simple too, if you want long running more complex microservices you're better off using ECS/EKS.

— nutrecht July 2018

“ “ The bigger problem with using spring (and java in general) is that lambda execution costs rise with memory usage. A lot of modern java frameworks assume that **memory is abundant** and cheap which isn't the case on lambda.

- davewritescode July 2018

“ “ The problem with java etc in a lambda is the **start up time**.

- moremattmattmatt July 2018

https://www.reddit.com/r/java/comments/8xv3ha/does_anybody_use_spring_boot_with_aws_lambda_do/

Meet Quarkus

Supersonic java



So who is this Quarkus— and what's all the fuss?



Up to 10x smaller, 100x faster
Seriously.



But wait, there's more!



Lower memory
usage



Faster startup



Optimized for
short-lived processes



Kubernetes
Native



developer
Enough ^ love to go around.



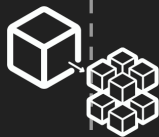
Live reload



Imperative and
reactive



Serverless



Microservices



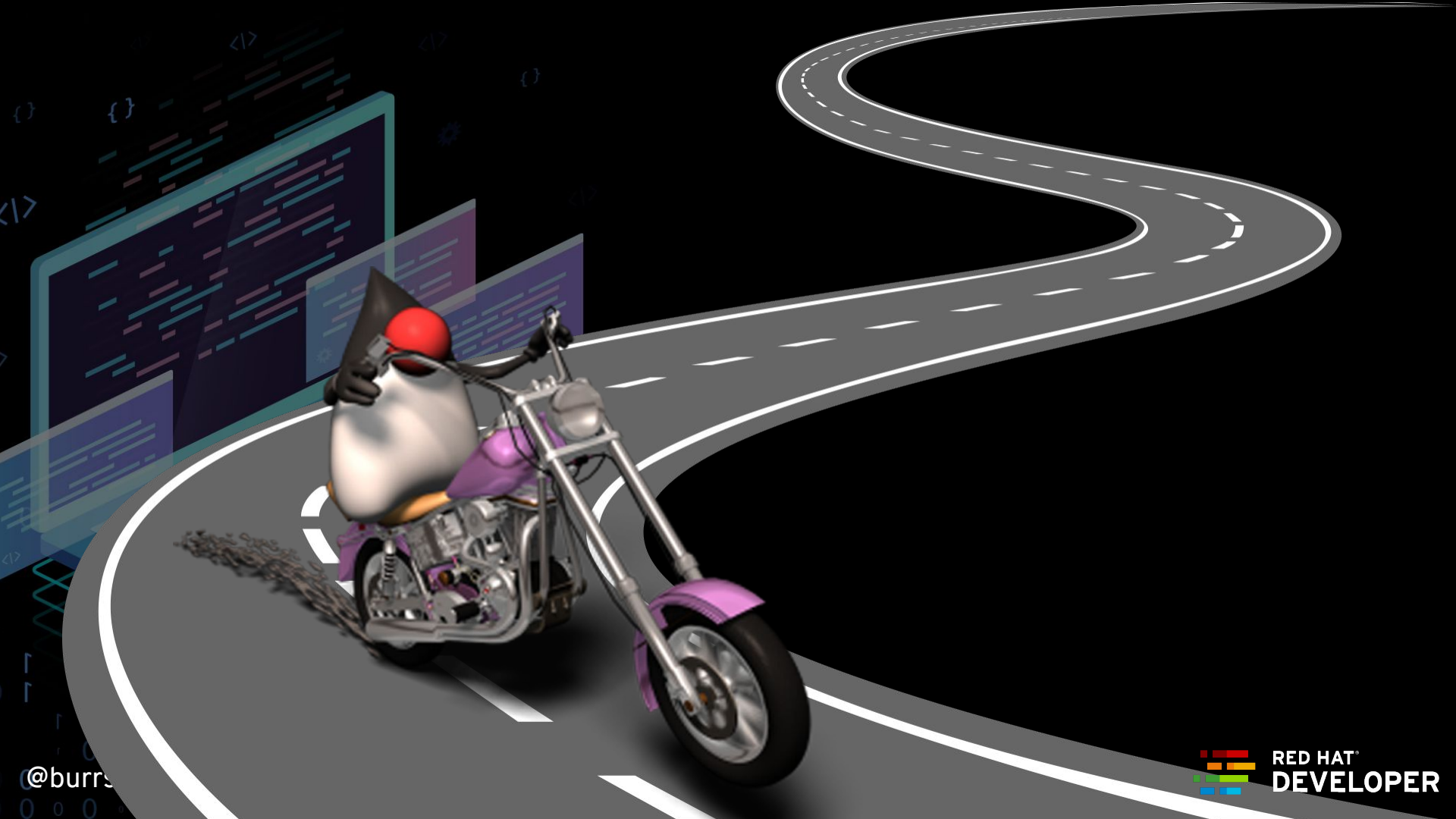
Fat jars
and native
executables



Optimized for
JAX-RS & JPA
(Hibernate) ...



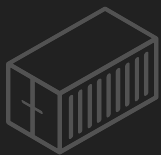
Kubernetes Native



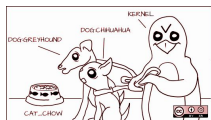
@burrs



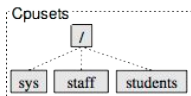




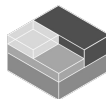
2000:
JAILS ADDED
TO FREEBSD



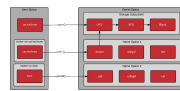
2003:
SELINUX ADDED TO
LINUX MAINLINE



2006:
GENERIC PROCESS
CONTAINERS



2008:
LINUX CONTAINER
PROJECT (LXC)



2008:
KERNEL AND USER
NAMESPACES



2013:
DOTCLOUD
BECOMES DOCKER



2013:
RED HAT
ENTERPRISE LINUX



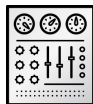
2015:
RHT CONTAINER
PLATFORM



2001:
LINUX -VSERVER
PROJECT



2005:
FULL RELEASE OF
SOLARIS ZONES



2007:
GPC RENAMED
CONTROL GROUPS



2013:
DOTCLOUD PYCON
LIGHTNING TALK



2014:
GOOGLE
KUBERNETES



2015:
STANDARDS VIA OCI
AND CNCF

2000

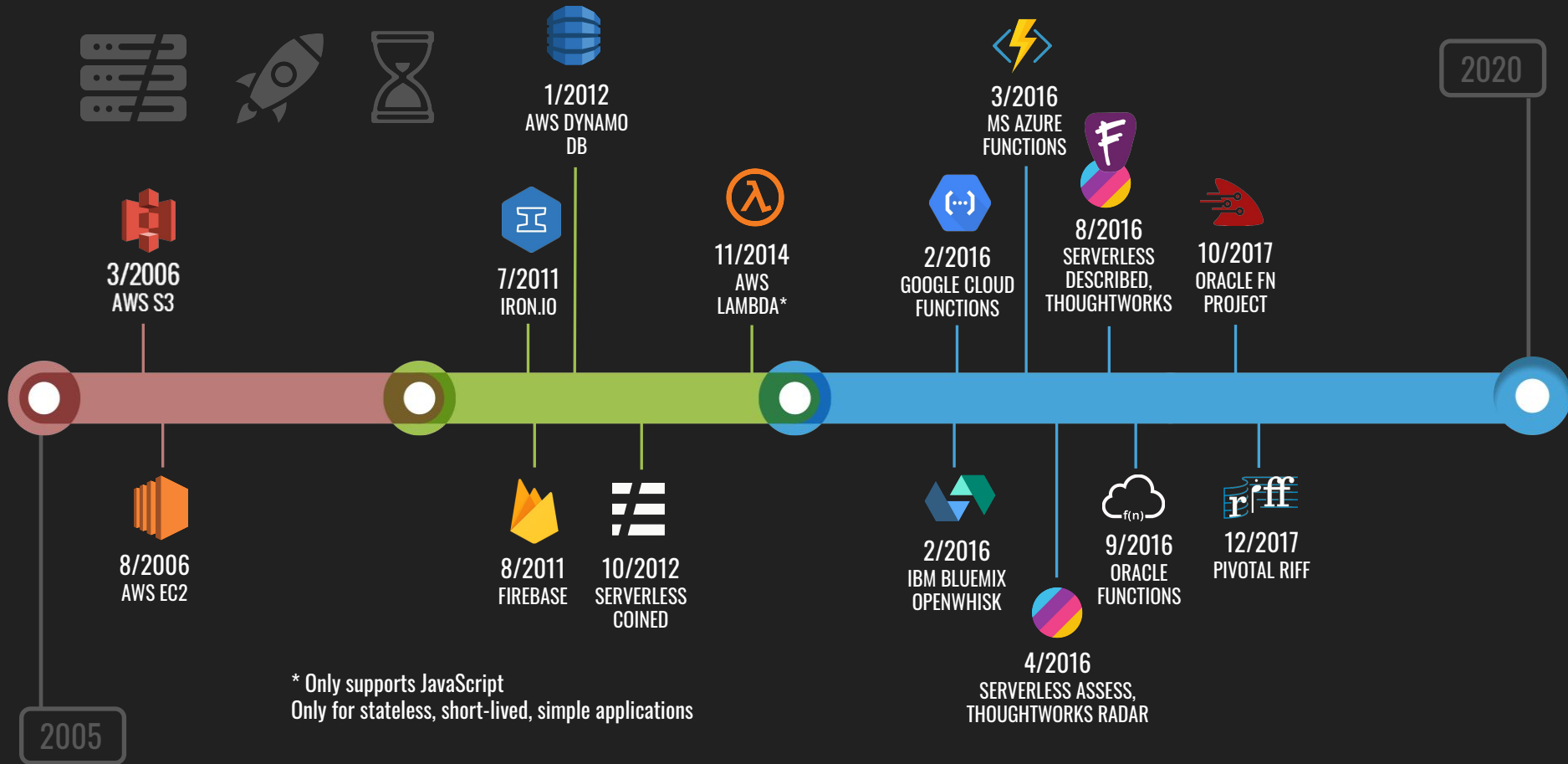
2005

2010

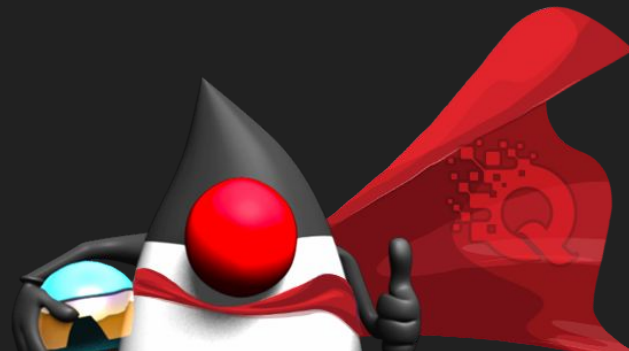
2015



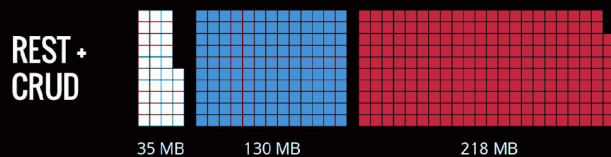








Memory (RSS) in Megabytes

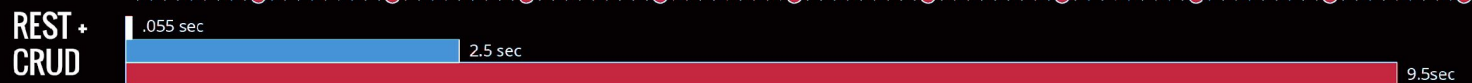


Quarkus with GraalVM

Quarkus with OpenJDK

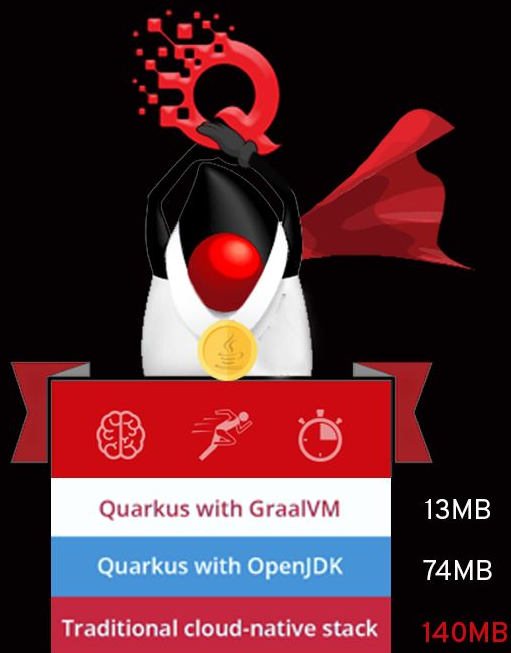
Traditional cloud-native stack

Boot + First Response Time in Seconds



MEMORY & BOOT + FIRST RESPONSE TIME

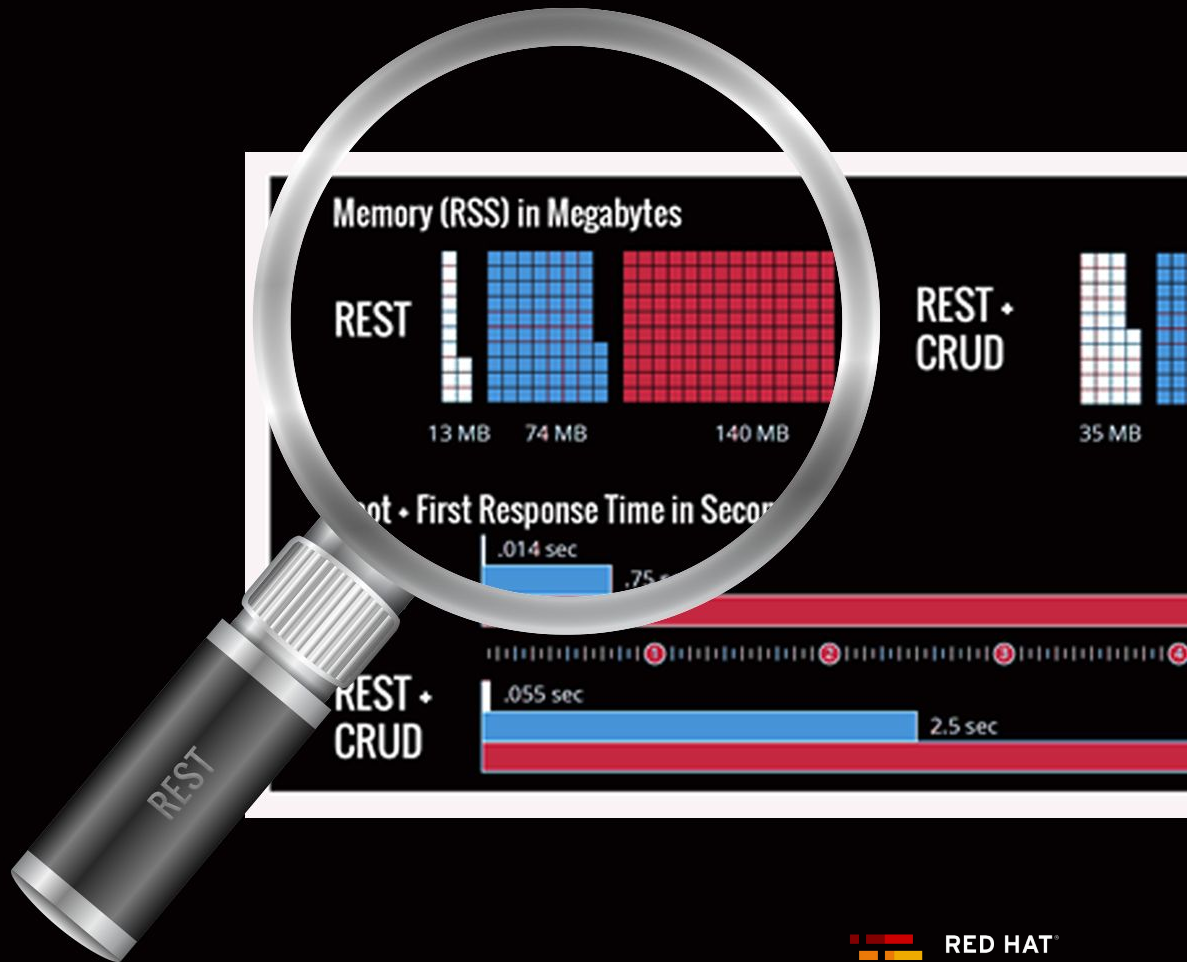


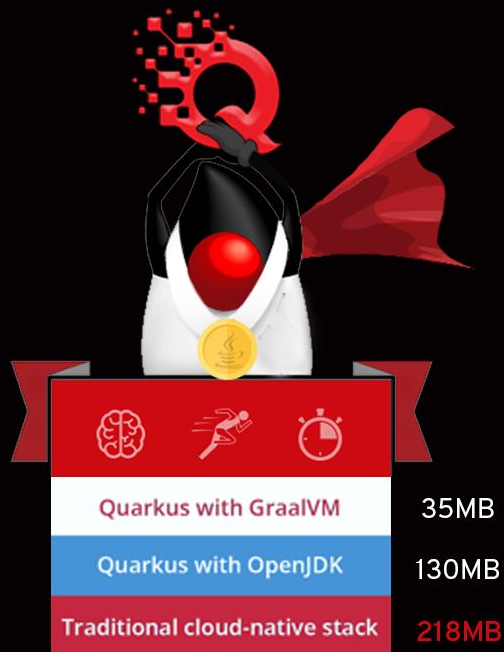


Memory (RSS) in Megabytes



@burrsutter bit.ly/quarkusjava

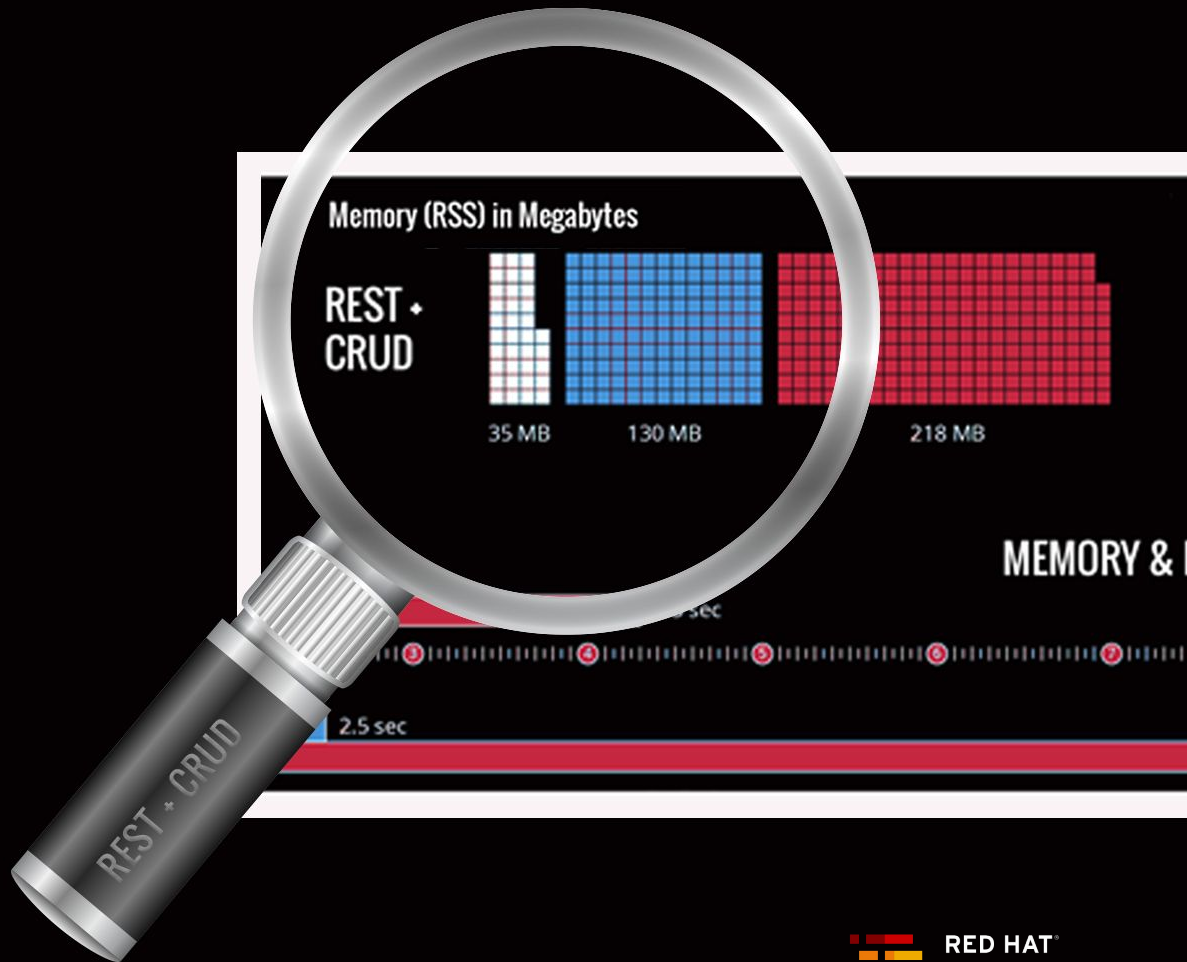




Memory (RSS) in Megabytes



@burrsutter bit.ly/quarkusjava

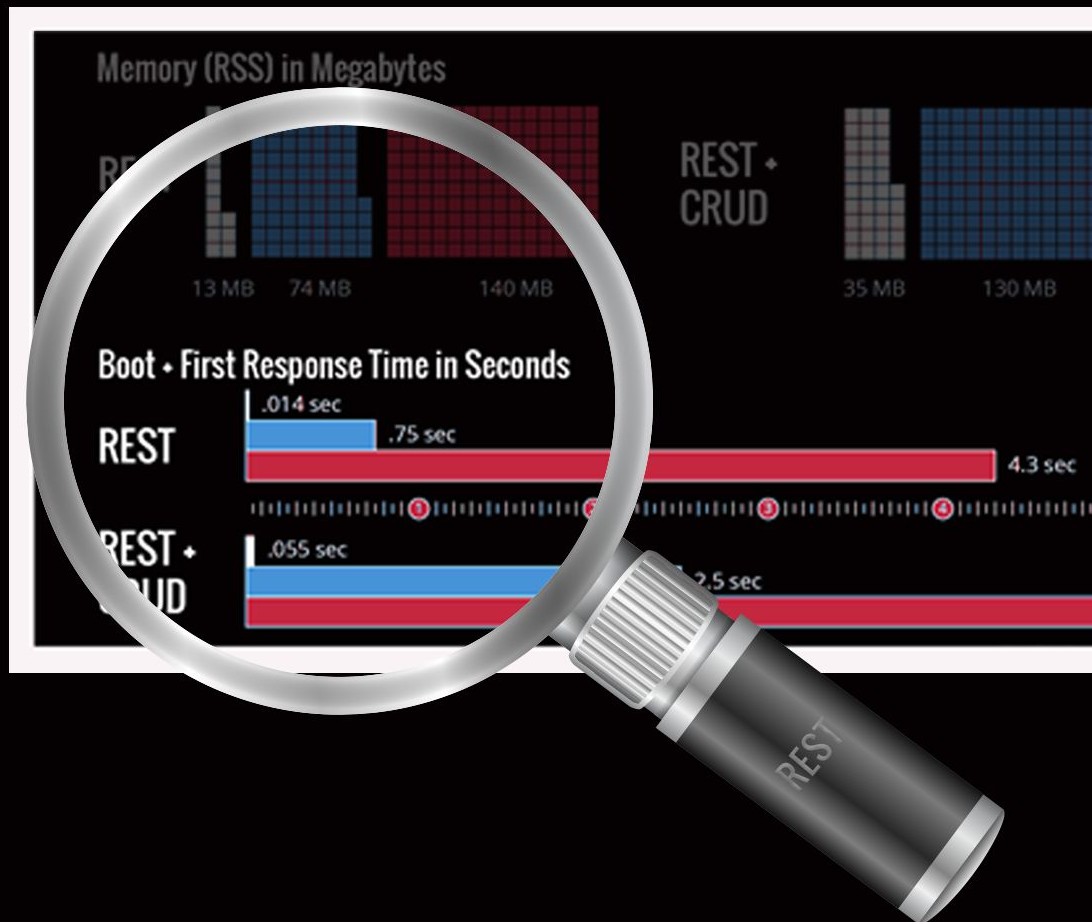




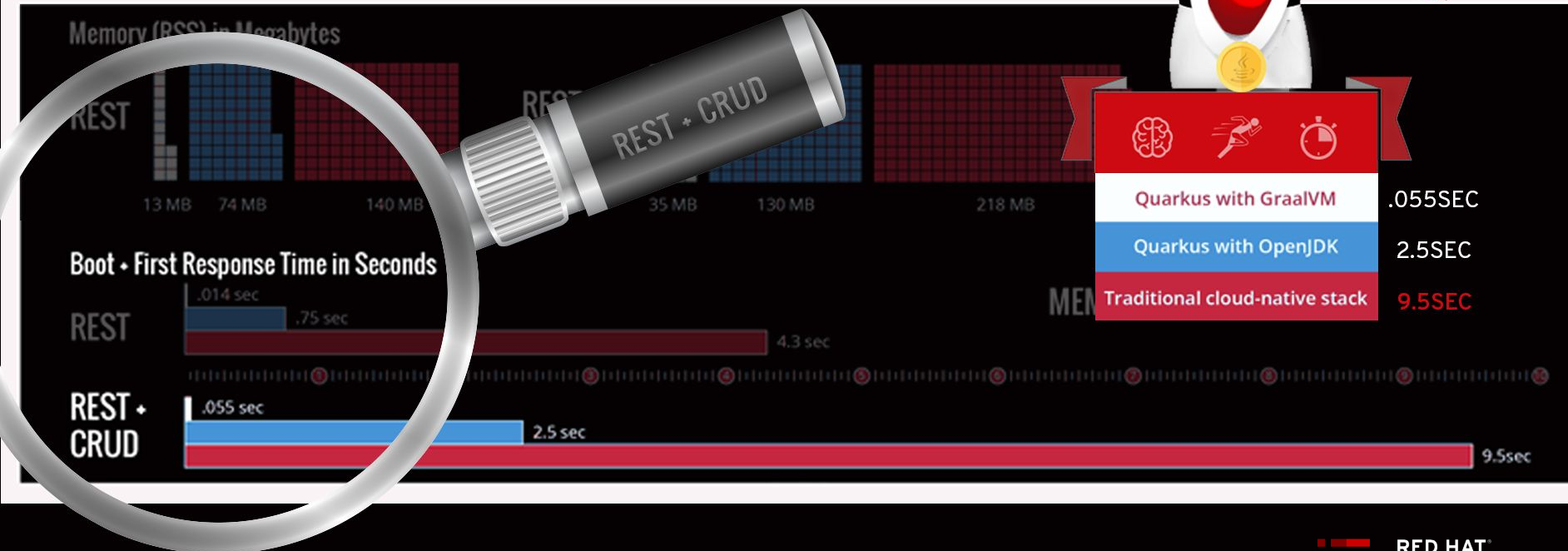
Boot + First Response Time in Seconds



@burrsutter bit.ly/quarkusjava



Boot + First Response Time in Seconds

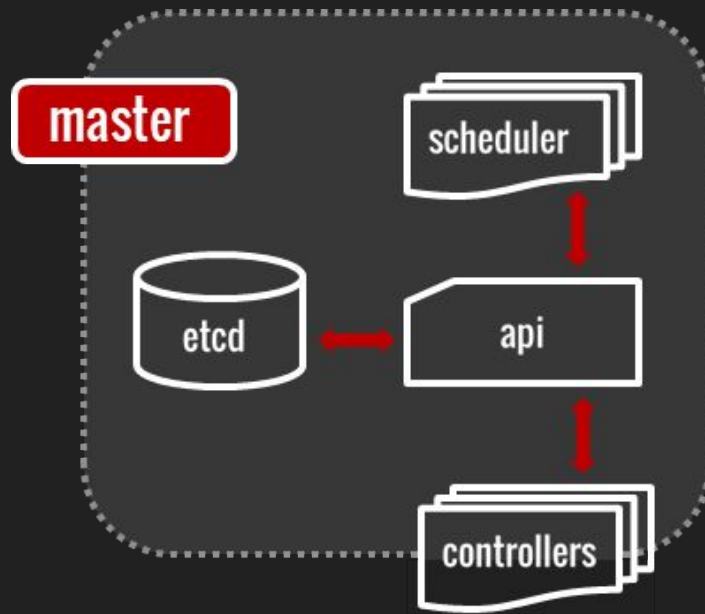


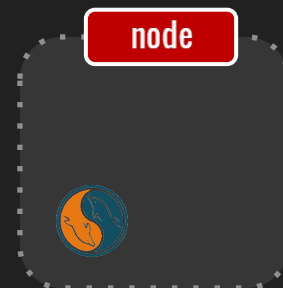
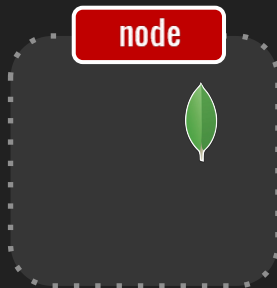
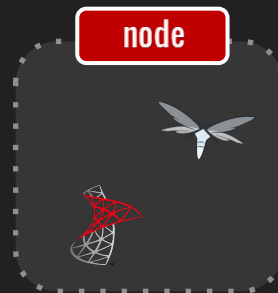
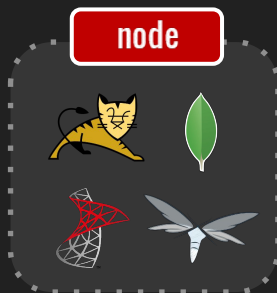
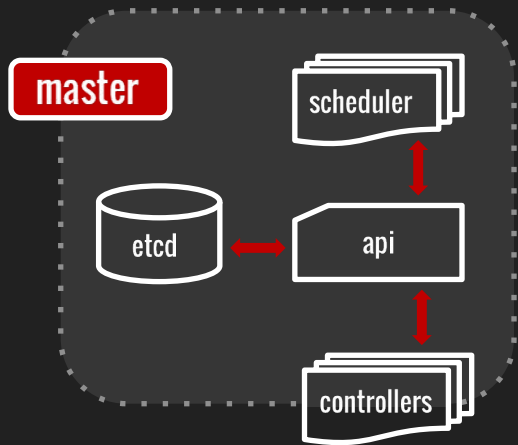
GraalVM™

Polyglot, Native or JVM, Embeddable

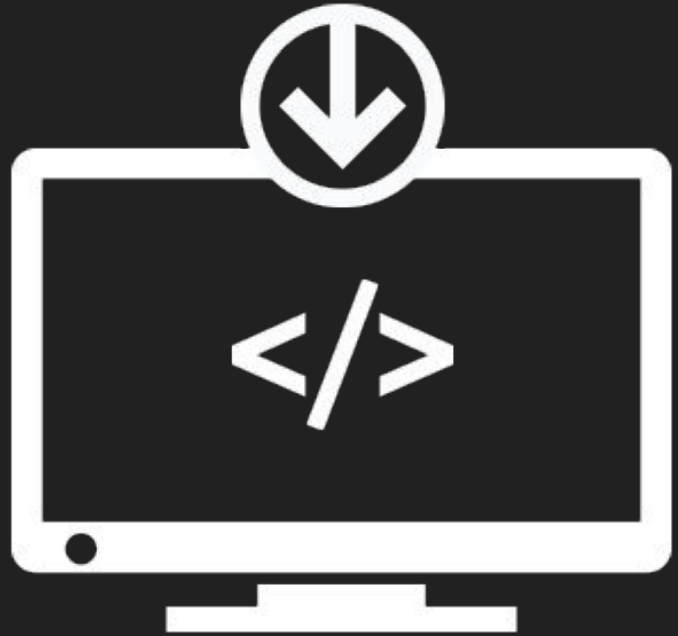
Kubernetes Orchestration



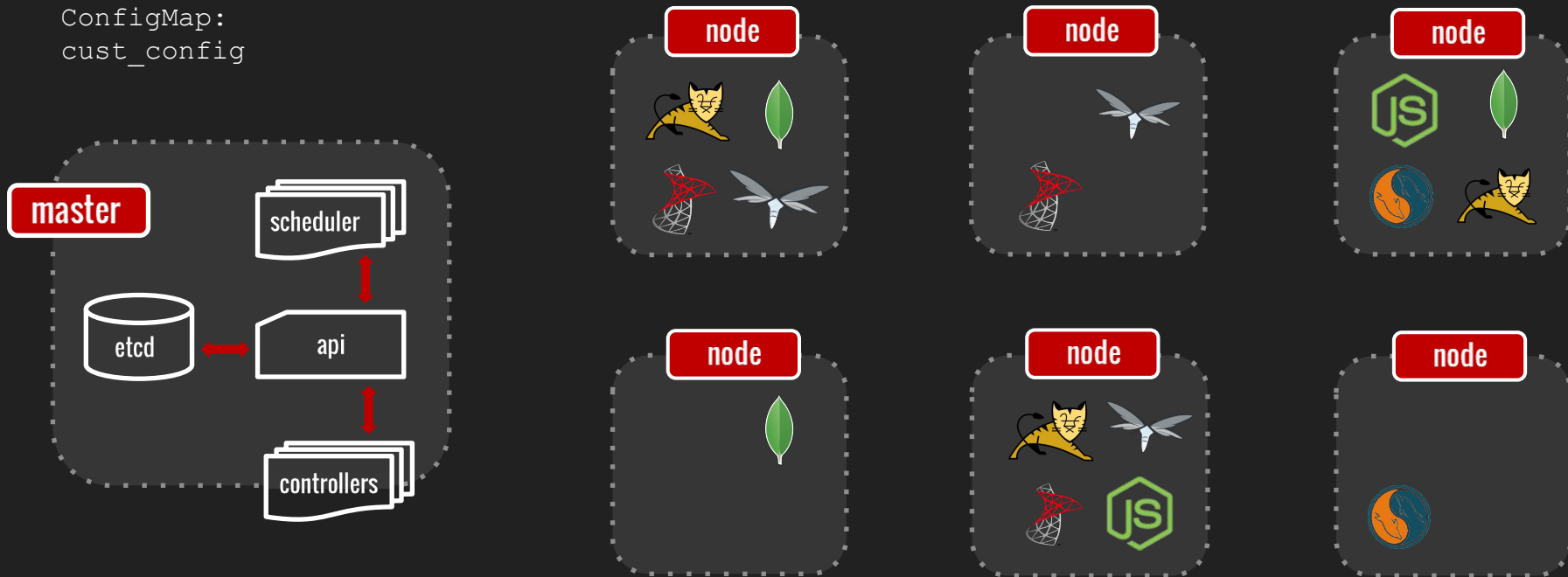




Declarative



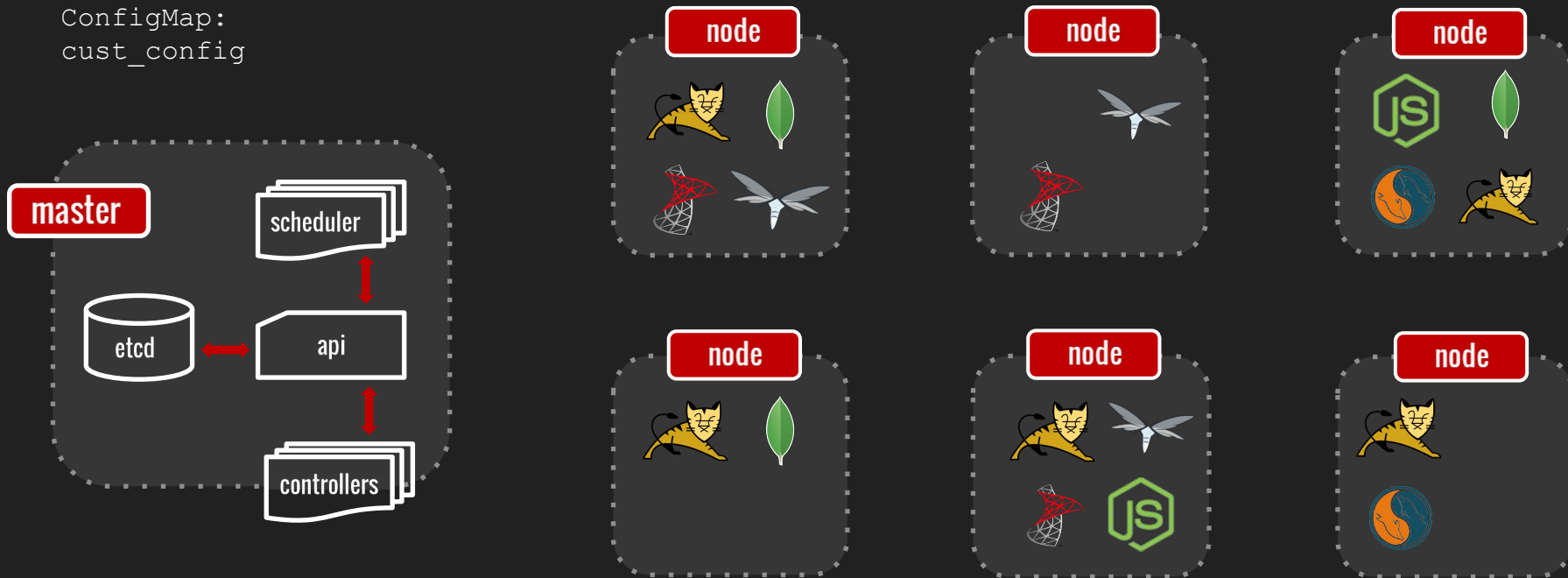
```
Image:
quay.io/images/custservice:1.1.0
Replicas:
2
Labels:
customerservice=prod,ci_build=1213
ConfigMap:
cust_config
```



Scheduled/Started



```
Image:
quay.io/images/custservice:1.1.0
Replicas:
2
Labels:
customerservice=prod,ci_build=1213
ConfigMap:
cust_config
```



Always Watching

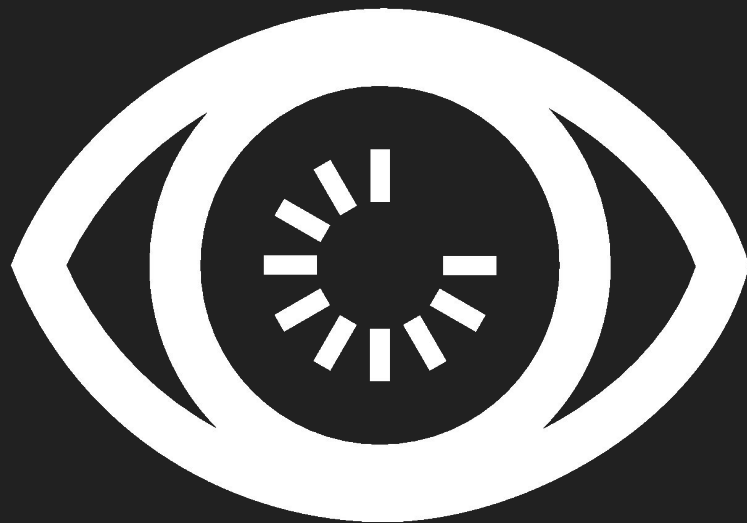
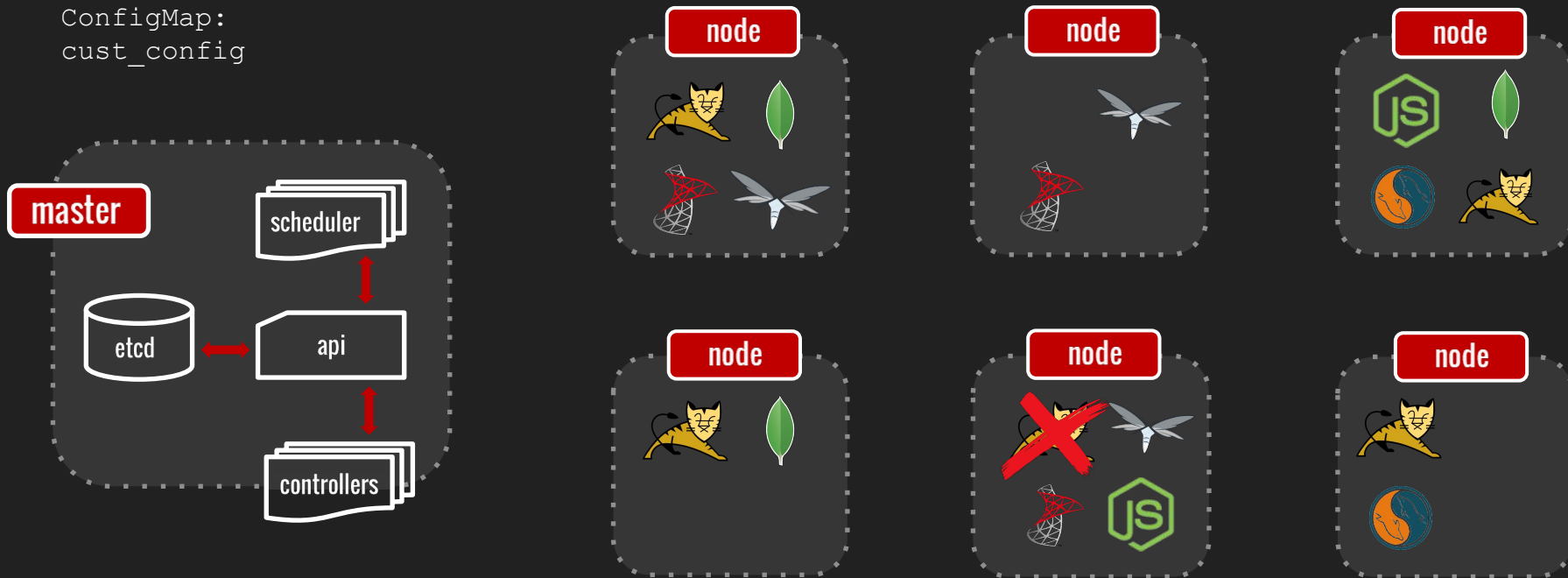


Image:
quay.io/images/custservice:1.1.0
Replicas:
2
Labels:
customerservice=prod,ci_build=1213
ConfigMap:
cust_config



Always Correcting

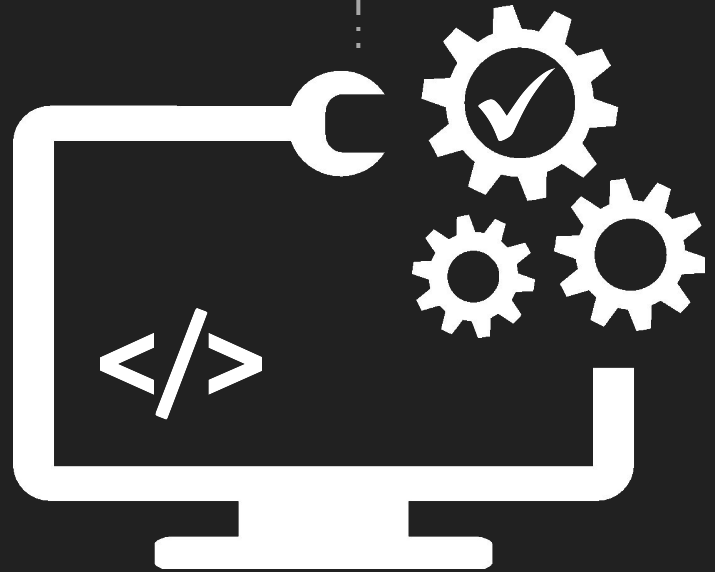
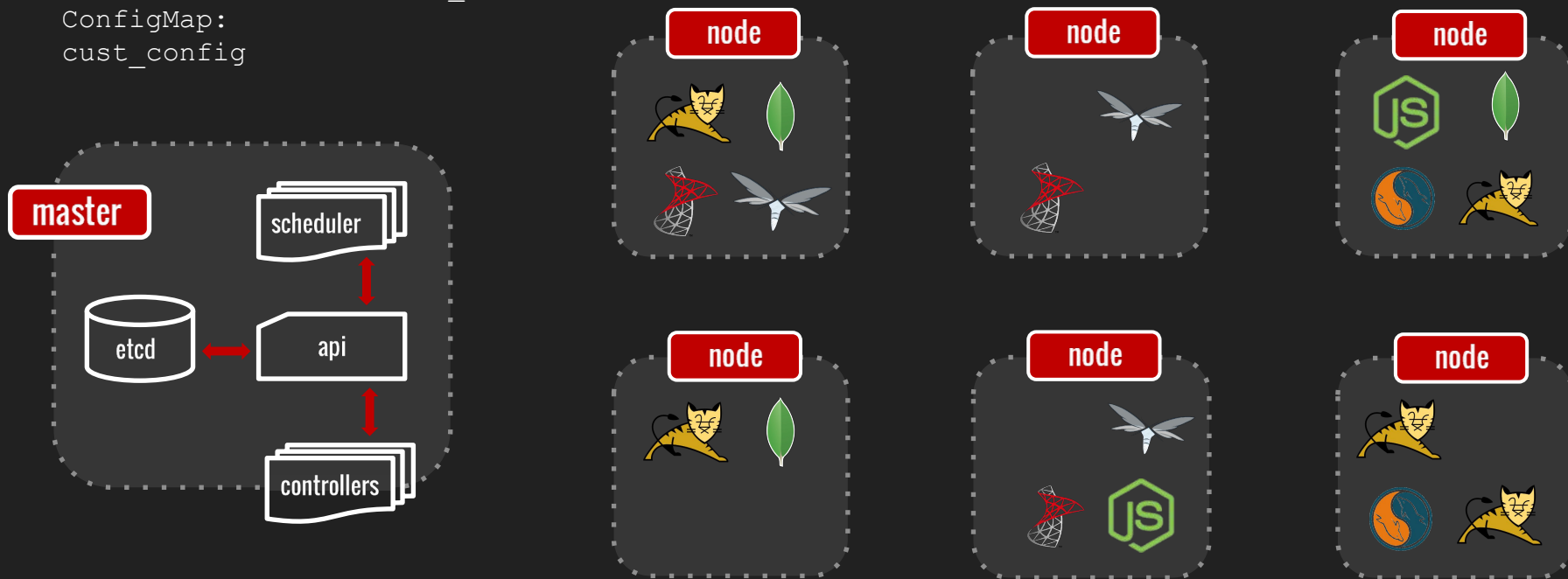


Image:
quay.io/images/custservice:1.1.0
Replicas:
2
Labels:
customerservice=prod,ci_build=1213
ConfigMap:
cust_config



Size and Speed Matter

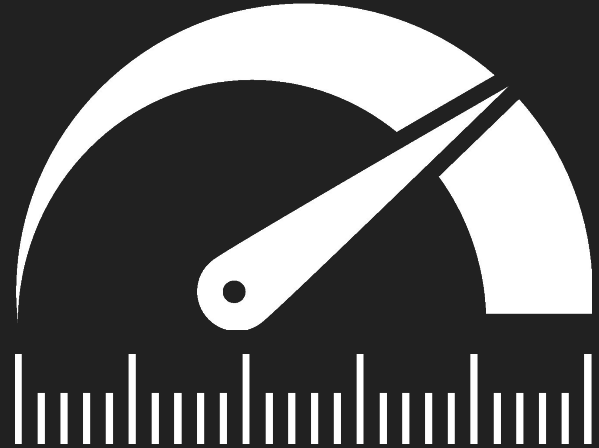
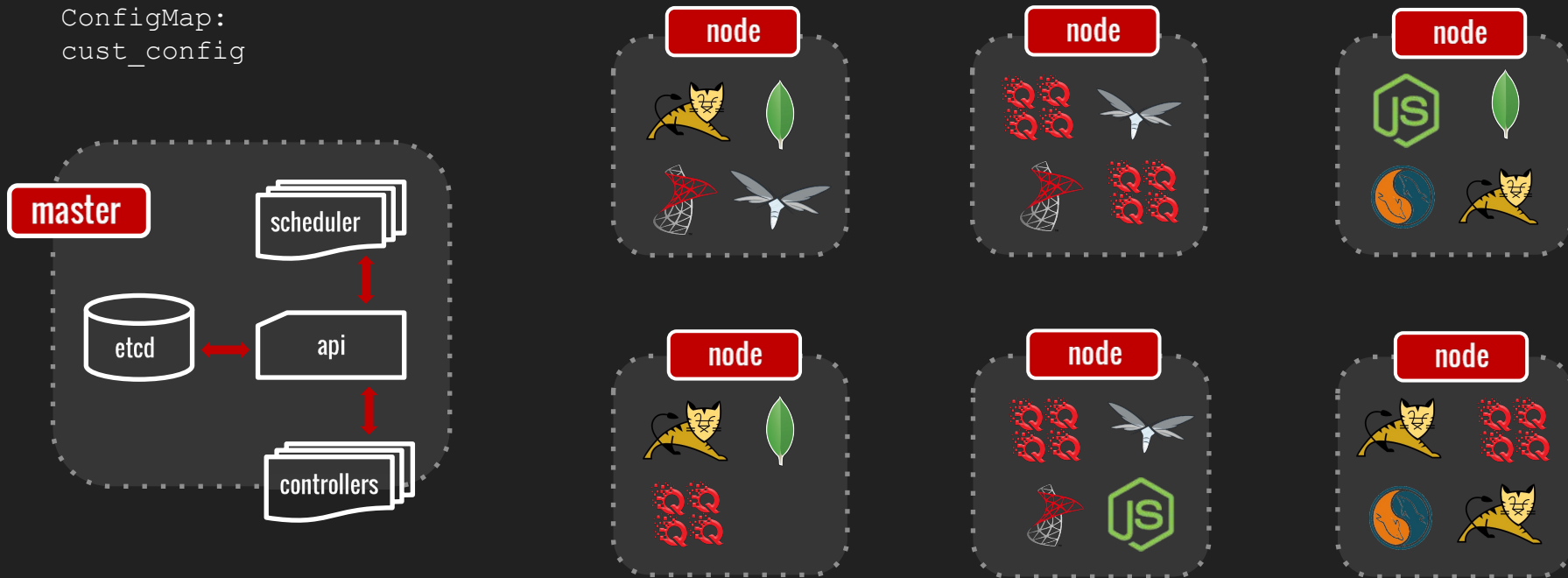
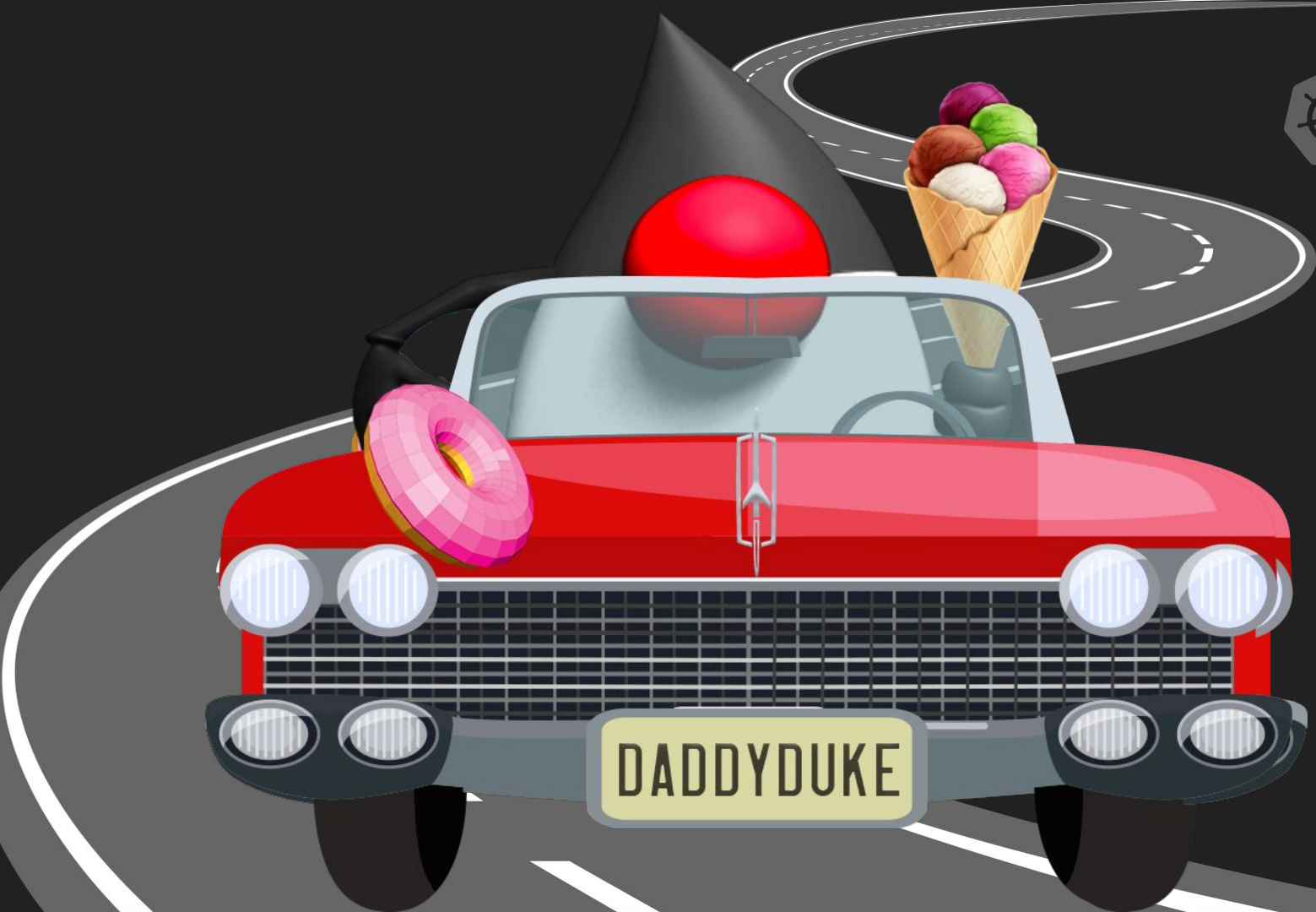


Image:
quay.io/images/custservice:1.1.0
Replicas:
20
Labels:
customerservice=prod,ci_build=1213
ConfigMap:
cust_config



Size and Speed Specs





RED HAT
DEVELOPER



10x smaller.

Up to 100x faster.

@burrsutter bit.ly/quarkusjava



RED HAT
DEVELOPER

It's just the
beginning ...



~~The End~~