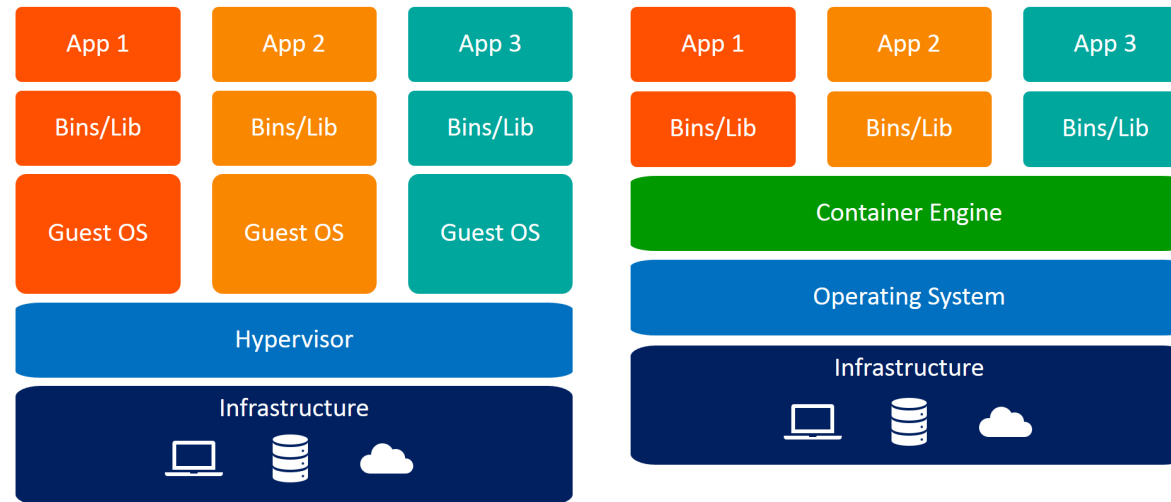


INTRO TO OPERATING SYSTEMS

VIRTUAL MACHINES/QUBES OS and
CONTAINERIZATION/ORCHESTRATION



THIS WEEK'S 2x75min 4p-5:15p

Virtual Machines Containers
<https://solutioncenter.apexsql.com/operating-system-os-performance-monitoring/>

- ◆ Emulation (5m)
- ◆ Virtual Machines (40m)

- ◆ Load Balancing (15m)
 - ◆ Containerization (20m)
 - ◆ Guest: Kubernetes/Docker/Rancher (60m)
-

◆ Emulation (5m)



<https://www.instructables.com/id/How-to-run-a-game-emulator/>



♦ Emulation (10m)

- ♣ Pick OS on Pr1mos: Like SQL server on Linux (mostly file system emu)
 - ♣ Early Retro Gaming: Timing Problems
 - ♣ DOSBox W3.1, W95 for current
 - ♣ Cygwin: Early Linux-on-Windows (NT, W95)
-

◆ Emulation (10m)

AlternativeTo
CROWDSOURCED SOFTWARE RECOMMENDATIONS

Find an alternative to... Find




 **DOSBox** 

DOSBox is a DOS-emulator that uses the SDL-library.

346 Like

Free Open Source Mac Windows Linux Android BSD FreeBSD

DOSBox is a DOS-emulator that uses the SDL-library. DOSBox emulates CPU:286/386 realmode/protected mode, Directory FileSystem/XMS/EMS, Tandy/Hercules/CGA/EGA/VGA/VESA graphics, a SoundBlaster/Gravis Ultra Sound card for excellent sound compatibility with older games...

◆ Virtual Machine (40m)

♣ Dual Boot is a Pain: Apple/Windows Linux/Windows (click)

♣ Configuration is a Pain: Tweaks, Libraries, Versions, Server Management (click)

♣ Reliability: Migrate OS Image to New Hardware (click)

♣ Cloning: Copy OS Image to New Hardware (click)

♣ StrongFencing/Sandboxing: Separate projects and users for security, SLA/performance (click)

<https://www.makeuseof.com/tag/reasons-dual-boot-linux/> https://en.wikipedia.org/wiki/Ubuntu_version_history <https://www.cloudwards.net/best-small-business-server/> <https://pcparamediccharlotte.com/tag/disk-cloning/> <https://www.slideshare.net/securityxploded/automating-malware-analysis>

♦ Virtual Machines ARE NOT:

- ♣ Virtual Reality

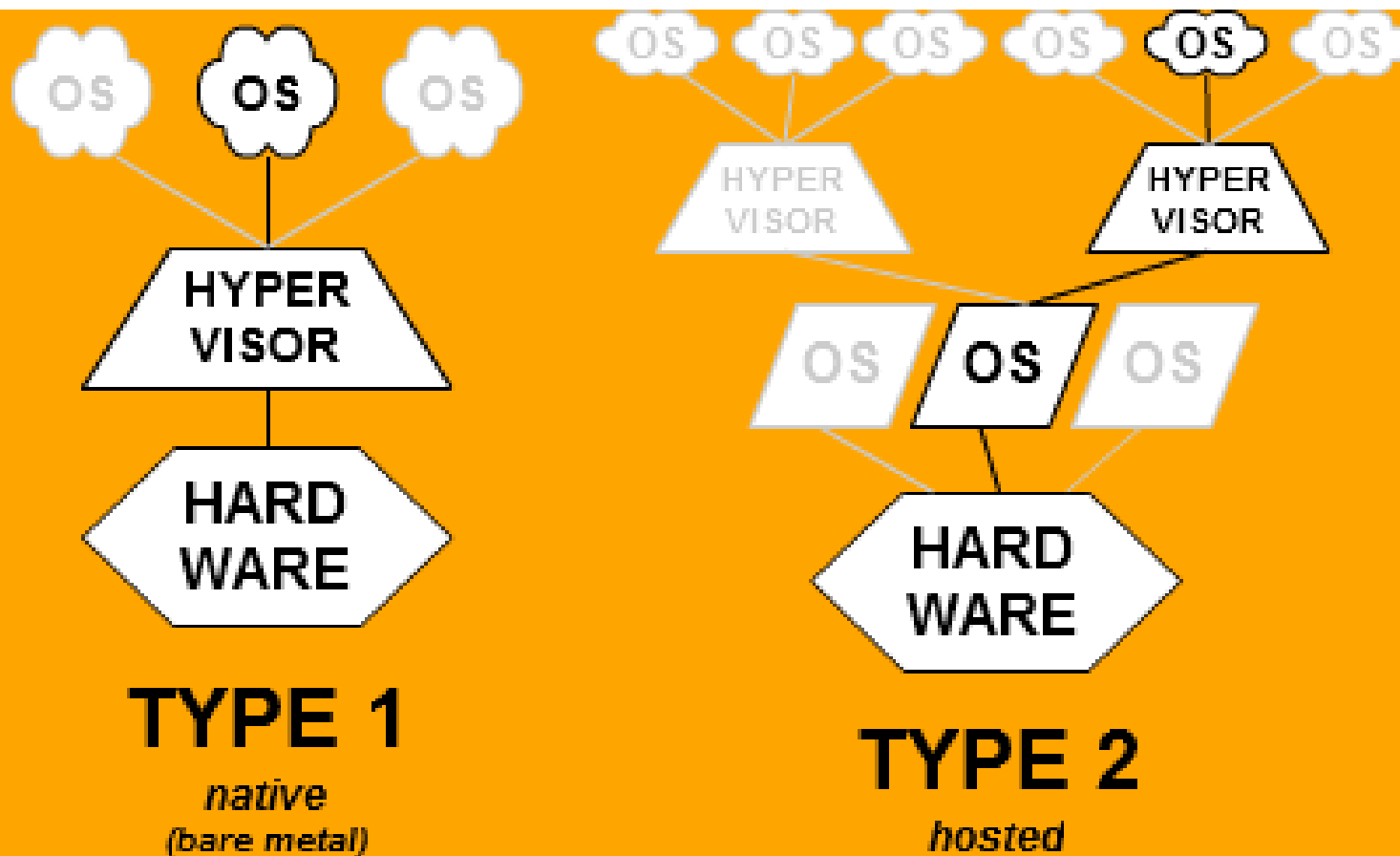
- ♣ Virtual Memory

- ♣ Virtual Desktop (similar)

- ♣ Java VM (AVM: Application Virtual Machine) (MRE: Managed Runtime Environment) (similar)

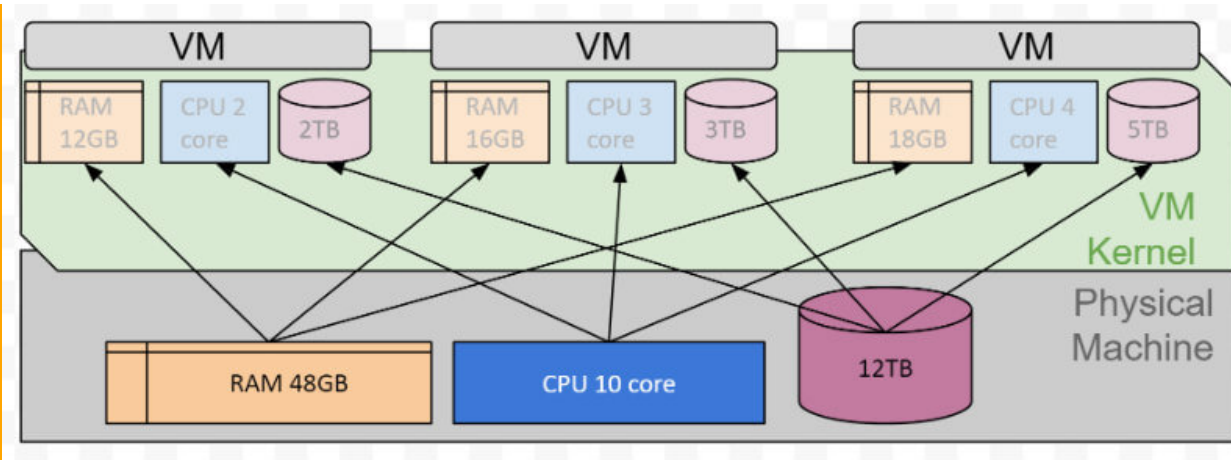
♦ Virtual Machines ARE:

♣ Usually Hypervisor-based Emulations of GUEST OS's on a HOST OS or BARE METAL



◆ Virtual Machines ARE:

♣ [Leading Type 1 Hypervisors](#): KVM, VMWARE ESXi, Xen, MSFT Hyper-V



Virtual machines MORE COMING SOON

Summary **Inventory**

[+ New](#) [Delete](#) [Start](#) [Turn off](#) [Shut down](#) [Save](#) [More](#) 20 items

| Name ↑ | State | Host server | CPU usage | Memory press... | Memory dema... | Assigned mem... | Uptime | Heartbeat | Protection... |
|--------------------------------|---------|-------------|-----------|-----------------|----------------|-----------------|------------|-----------|---------------|
| VM-NY-SEARCH1 | Running | 43C07-27 | 0% | 75% | 364 MB | 512 MB | 4:12:41:09 | OK | Not protected |
| VM-NY-SEARCH2 | Running | 43C07-26 | 0% | 71% | 363 MB | 512 MB | 4:02:39:51 | OK | Not protected |
| VM-NY-SEARCH3 | Running | 43C07-27 | 0% | 73% | 373 MB | 512 MB | 4:09:47:45 | OK | Not protected |
| VM-NY-SQL1 | Running | 43C07-26 | 0% | 72% | 368 MB | 512 MB | 4:02:27:14 | OK | Not protected |
| VM-NY-SQL3 | Running | 43C07-27 | 1% | 83% | 730 MB | 680 MB | 4:12:23:58 | OK | Not protected |
| VM-NY-WEB1 | Running | 43C07-24 | 0% | 73% | 373 MB | 512 MB | 4:02:06:55 | OK | Not protected |
| VM-NY-WEB2 | Running | 43C07-26 | 0% | 73% | 373 MB | 512 MB | 4:02:08:55 | OK | Not protected |
| VM-NY-WEB3 | Running | 43C07-26 | 0% | 73% | 373 MB | 512 MB | 4:02:03:03 | OK | Not protected |
| VM-NY-WEB4 | Running | 43C07-26 | 0% | 73% | 373 MB | 512 MB | 4:01:59:03 | OK | Not protected |
| VM-SEA-SEARCH1 | Running | 43C07-24 | 0% | 72% | 368 MB | 512 MB | 4:02:27:41 | OK | Not protected |
| VM-SEA-SEARCH2 | Running | 43C07-27 | 0% | 69% | 353 MB | 512 MB | 4:01:23:11 | OK | Not protected |
| VM-SEA-SEARCH3 | Running | 43C07-24 | 0% | 75% | 364 MB | 512 MB | 4:02:21:54 | OK | Not protected |
| VM-SEA-WEB1 | Running | 43C07-26 | 0% | 71% | 363 MB | 512 MB | 3:18:50:04 | OK | Not protected |
| VM-SEA-WEB2 | Running | 43C07-27 | 0% | 73% | 373 MB | 512 MB | 3:19:09:07 | OK | Not protected |
| VM-SEA-WEB3 | Running | 43C07-24 | 0% | 74% | 378 MB | 512 MB | 4:02:09:56 | OK | Not protected |
| VM-SEA-WEB4 | Running | 43C07-27 | 0% | 53% | 271 MB | 512 MB | 4:10:50:15 | OK | Not protected |
| VM-SEA-WEB5 | Running | 43C07-27 | 0% | 51% | 261 MB | 512 MB | 4:10:36:44 | OK | Not protected |

<https://actusdigital.com/virtual-machine-vs-physical-server/> <https://docs.microsoft.com/en-us/windows-server/manage/windows-admin-center/use/manage-virtual-machines>

◆ Virtual Machines ARE:

♣ Usually for Cloud Providers to Make Better Use of Big Servers

- *Esp. for single-threaded single-core apps on multi-core CPUs*
- *Esp. for bursty requirements per user*
- *Esp. when remote infrastructure ok or even required*

♣ Usually for Cloud Users to Have Scalability, Reliability, Shared Costs

- *Esp. when performance is not the only consideration*

- *Esp. when keeping machines patched and connected*
- *Esp. when lots of small-transaction, global customer traffic, MicroServices*

♦ Virtual Machines ARE:

- ♣ The Killer Solution for Malware? Open Everything in a New VM!
- ♣ The Modern Mainframe? The Shared SuperComputer? Global TimeSharing? Back to Batch?

- ♣ The New Professional Computing Environment, Like it or Not

QubesOS

Rutkowska

- ♣ Interesting from a Performance Pt of View
- *Was slow, no longer so*

- 5-10% slow-down for cpu, more for disk/net i/o
[VMWARE 2007](#) [BARE METAL 2018](#) [REDDIT 2019](#)

- Each VM uses 300-500MB of memory by itself
- More jobs per machine because of overlapping idle

♦ **Virtual Machines ARE NOT BAD**
Memory/Disk mapped but not allocated (eager vs lazy/thick thin provisioning even compressed)

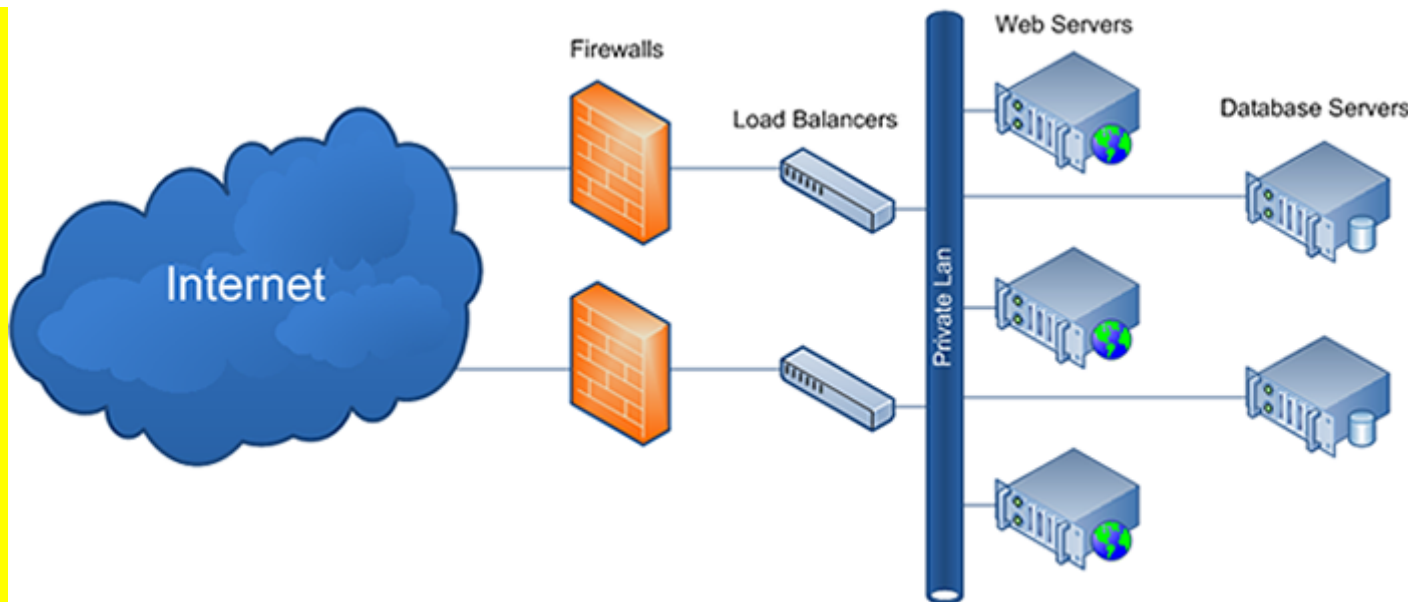
♣ **RENEW WAY** to maximize/manage resource utilization

| Type | CPU test | Memory test | Disk test | Cinebench |
|------------|----------|-------------|-----------|-----------|
| Bare Metal | -0% | -0% | -0% | -0% |

| | | | | |
|--------------------|------|------|------|-----|
| Bhyve | -19% | -63% | -25% | -8% |
| ESXi | -1% | -4% | -30% | +1% |
| HyperV | -6% | -18% | -1% | -2% |
| KVM/KVMCPU/IDE | -10% | -1% | -50% | -2% |
| KVM/HostCPU/VirtIO | -1% | -4% | -1% | -2% |

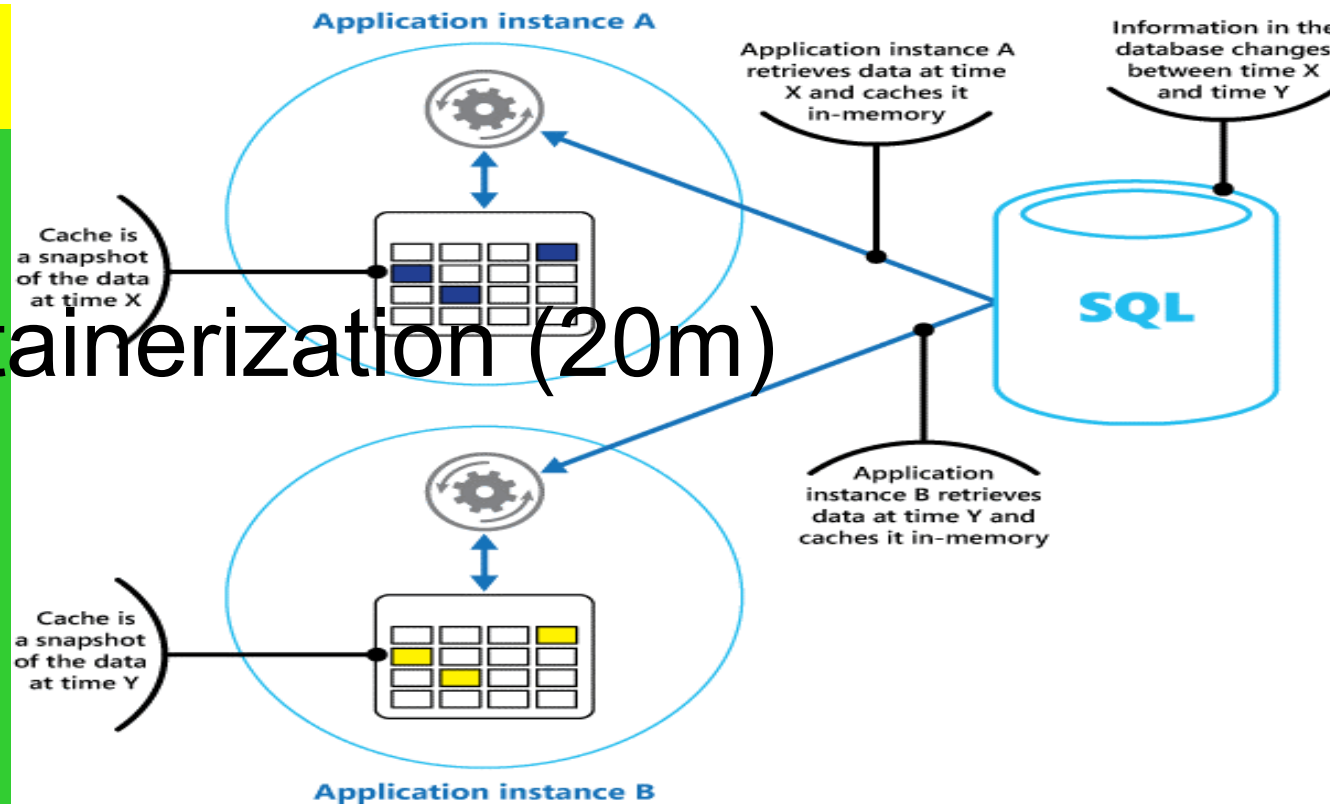
◆ Load Balancing (5m)

♣ Server Load Balancing IS:



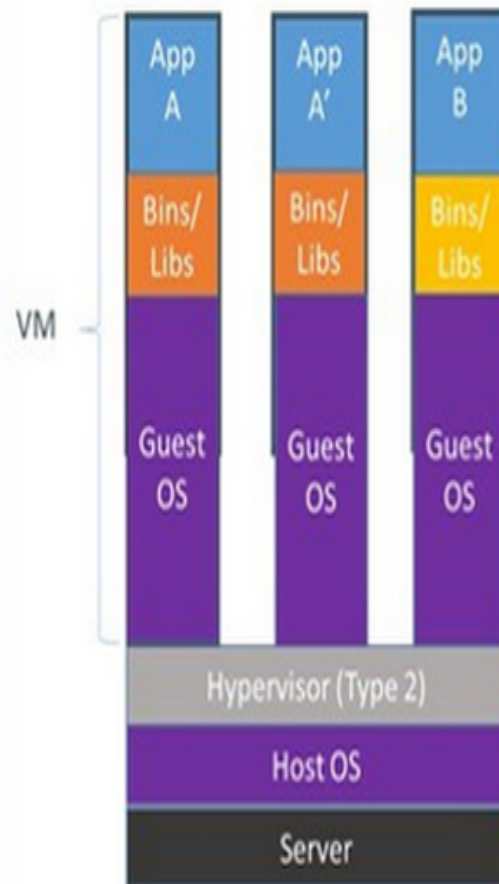
- ♣ Traditionally, for multiple web/sql servers w/millions of users
- ♣ Round robin, random, fewest connx, least load, hashed, sticky

◆ Containerization (20m)

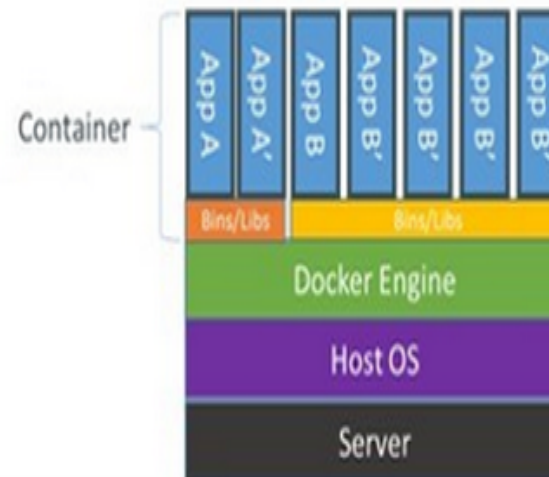


<https://www.idaq.com/info/load-balanced-dedicated-servers> <https://docs.microsoft.com/en-us/azure/architecture/best-practices/caching>

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



♣ Contain

♣ Docker

♣ Dispos
Deployable

```
Prachis-MBP:letschat prachi$ cat ./docker-compose.yml
version: '2'
services: letschat prachi$
  mongo:
    image: mongo
    stdin_open: true
    tty: true
    ports:
      - "27017"
    labels:
      io.rancher.container.pull_image: always
  chat:
    image: sdelements/lets-chat
    stdin_open: true
    tty: true
    ports:
      - 9890:8080/tcp
    labels:
      io.rancher.container.pull_image: always
```

```
Prachis-MBP:letschat prachi$ cat ./chat-deployment.yml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    io.rancher.container.pull_image: always
  creationTimestamp: null
  labels:
    io.kompose.service: chat
  name: chat
spec:
  replicas: 1
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        io.kompose.service: chat
    spec:
      containers:
        - image: sdelements/lets-chat
          name: chat
          ports:
            - containerPort: 8080
          resources: {}
          stdin: true
          tty: true
          restartPolicy: Always
      status: {}
```

```
Prachis-MBP:letschat prachi$ cat ./chat-service.yml
apiVersion: v1
kind: Service
metadata:
  annotations:
    io.rancher.container.pull_image: always
  creationTimestamp: null
  labels:
    io.kompose.service: chat
  name: chat
spec:
  ports:
    - name: "9890"
      port: 9890
      targetPort: 8080
  selector:
    io.kompose.service: chat
status:
  loadBalancer: {}
```

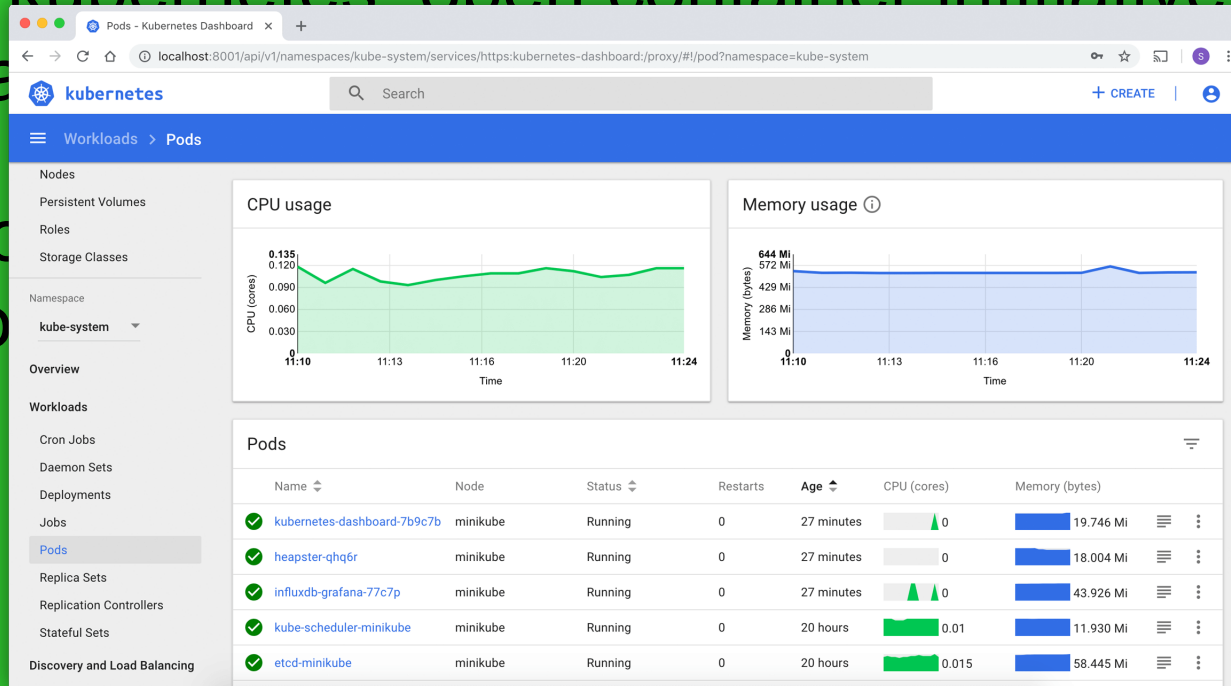
me

Created by Paint X

<https://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/> <https://rancher.com/blog/2018/2018-08-02-journey-from-cattle-to-k8s/>

♣ docker, kubernetes, open container initiative, coreos/rkt, apache

♣ cluster of nodes (an app), a pod



<https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>

♣ Spinning up or down a pod is 5-15sec

♣ Spinning up or down a vm is 1-10min

♣ Containers include libraries minimized for App, BUT NOT other OS parts

♣ Containers are [security--] while VMs are [security++]

<https://stackoverflow.com/questions/30869185/kubernetes-pod-creation-speed>

