KubeCon | CloudNativeCon

Europe 2019

Kubernetes + Encrypted Memory = Security * Privacy

Disclaimer

Please Note:

- IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM’s sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Who Are We?

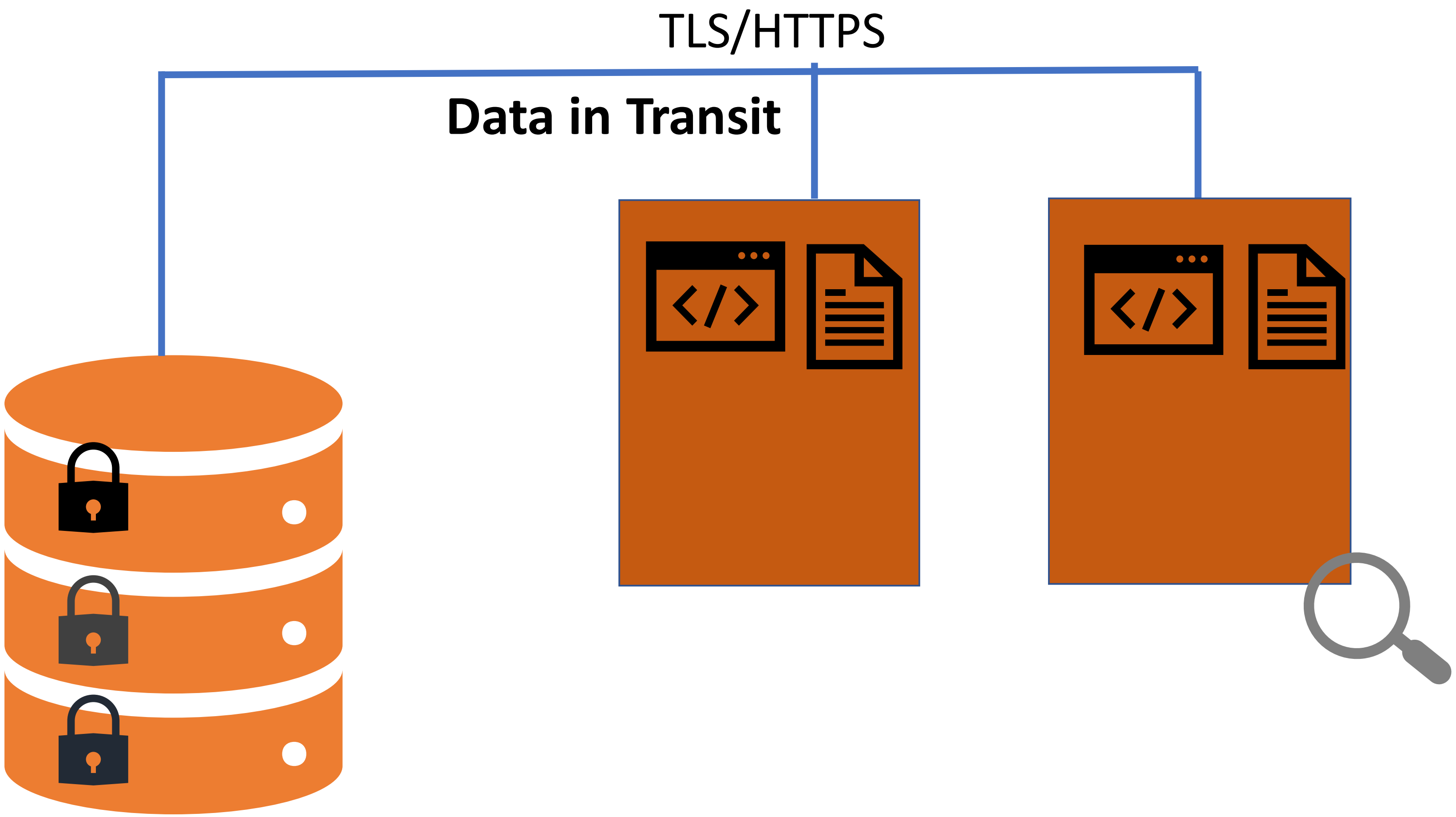
Harshal Patil

harche

Pradipta Banerjee

@pradipta_kr

How Do We Secure Data and Code?



Data in Transit

Data at Rest

Data in Use ??

- From other software
- Malicious Admins
- Compromised host/hypervisor

What is being done?

IBM Power

- Secure VM and Protected Execution Facility (PEF)

Intel

- SGX
- Total Memory Encryption - TME/MKTME

AMD

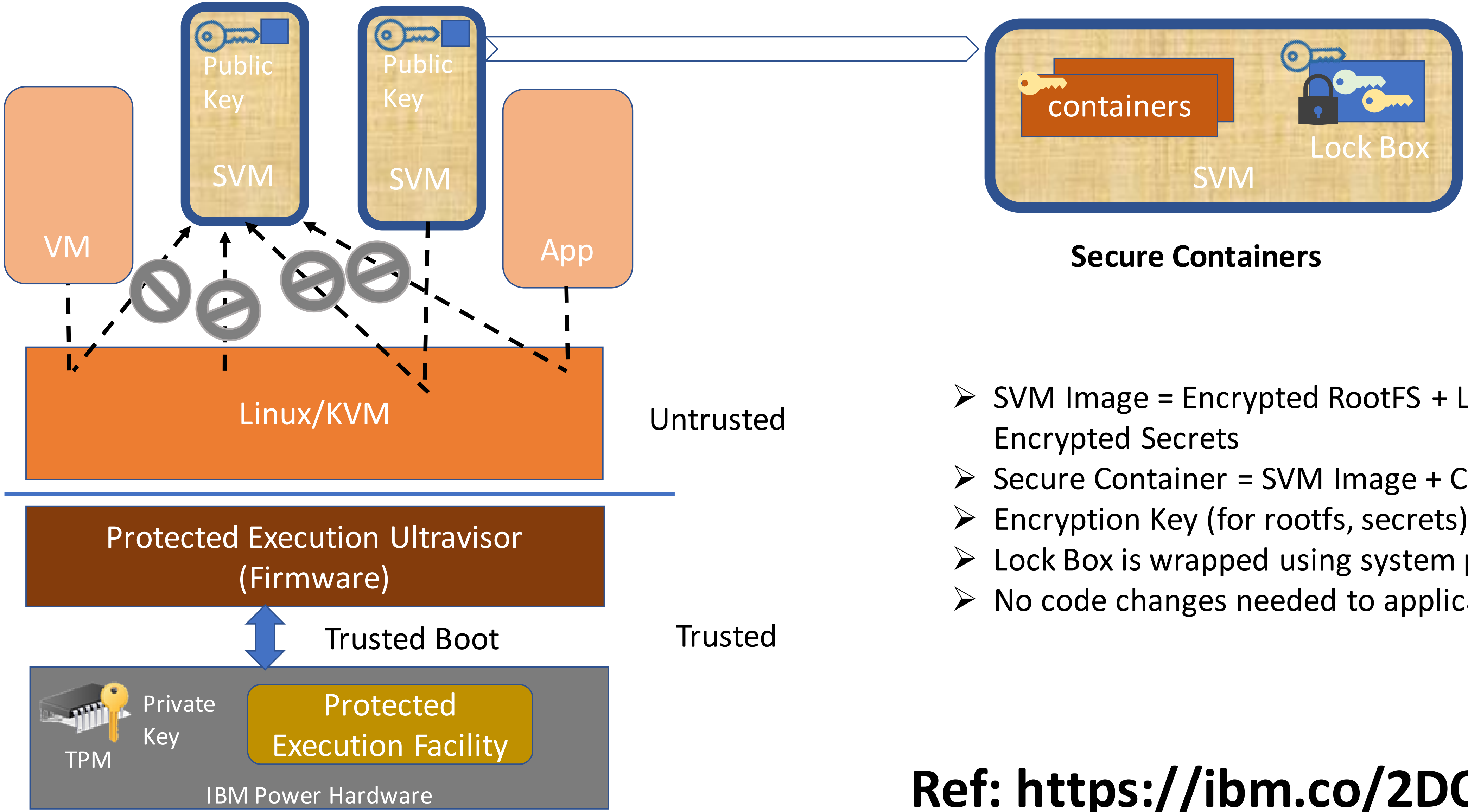
- Secure Memory Encryption
- Secure Encrypted Virtualization (SEV)

Explain Like I'm 5

Create a black box

- Stuffs inside the black box is protected from anything that is outside.

Secure VM and Secure Containers



Secure Containers

- ➢ SVM Image = Encrypted RootFS + Lock Boxes + Encrypted Secrets
- ➢ Secure Container = SVM Image + Container RootFS
- ➢ Encryption Key (for rootfs, secrets) put in Lock Box
- ➢ Lock Box is wrapped using system public key
- ➢ No code changes needed to application or container

Ref: https://ibm.co/2DOL7LJ

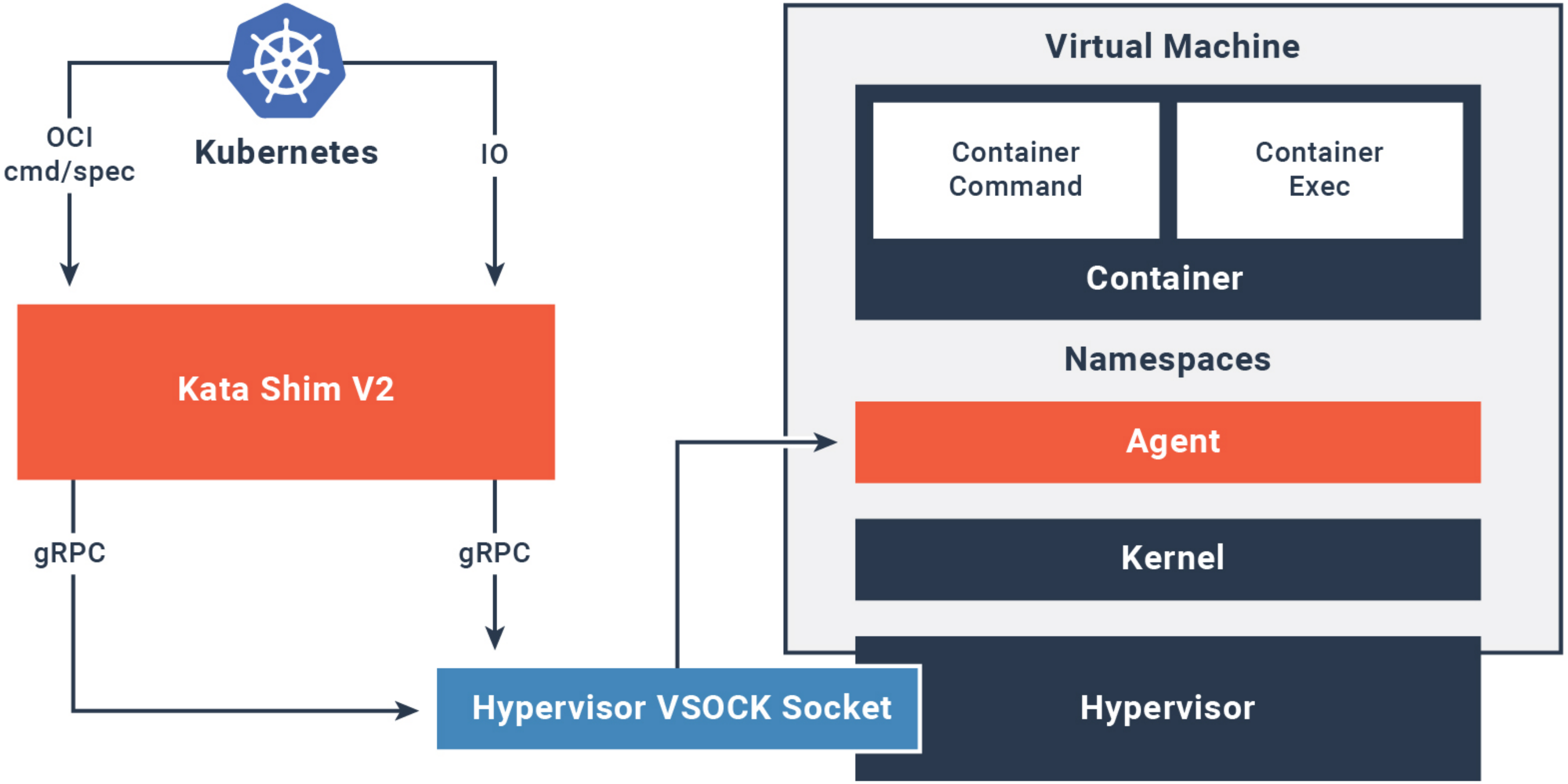
How can we use it with k8s?

Leveraging Kata container runtime

- Kata launches Secure containers (SVM + container)
- Since the memory pages of the Secure VM are protected so are the memory pages of the containers running inside those Secure VMs

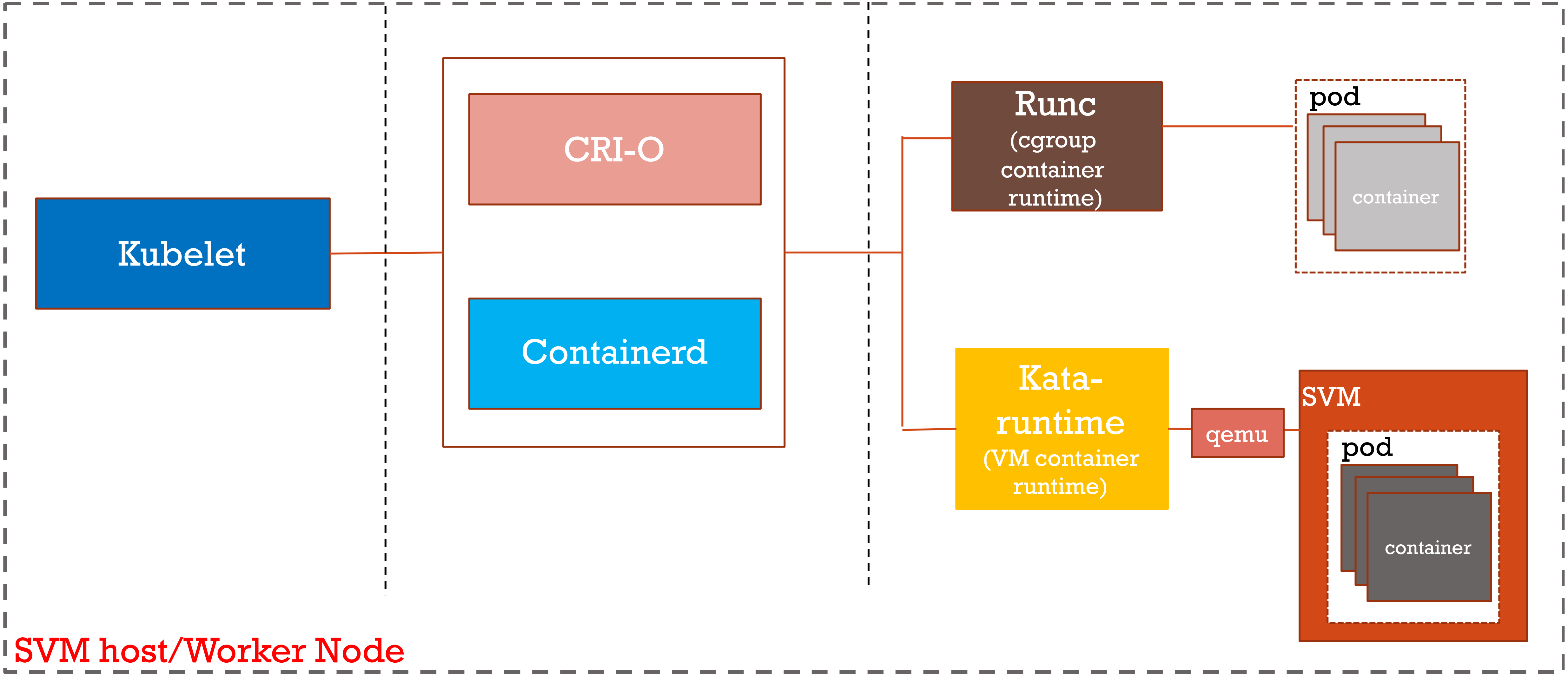
Kata Containers

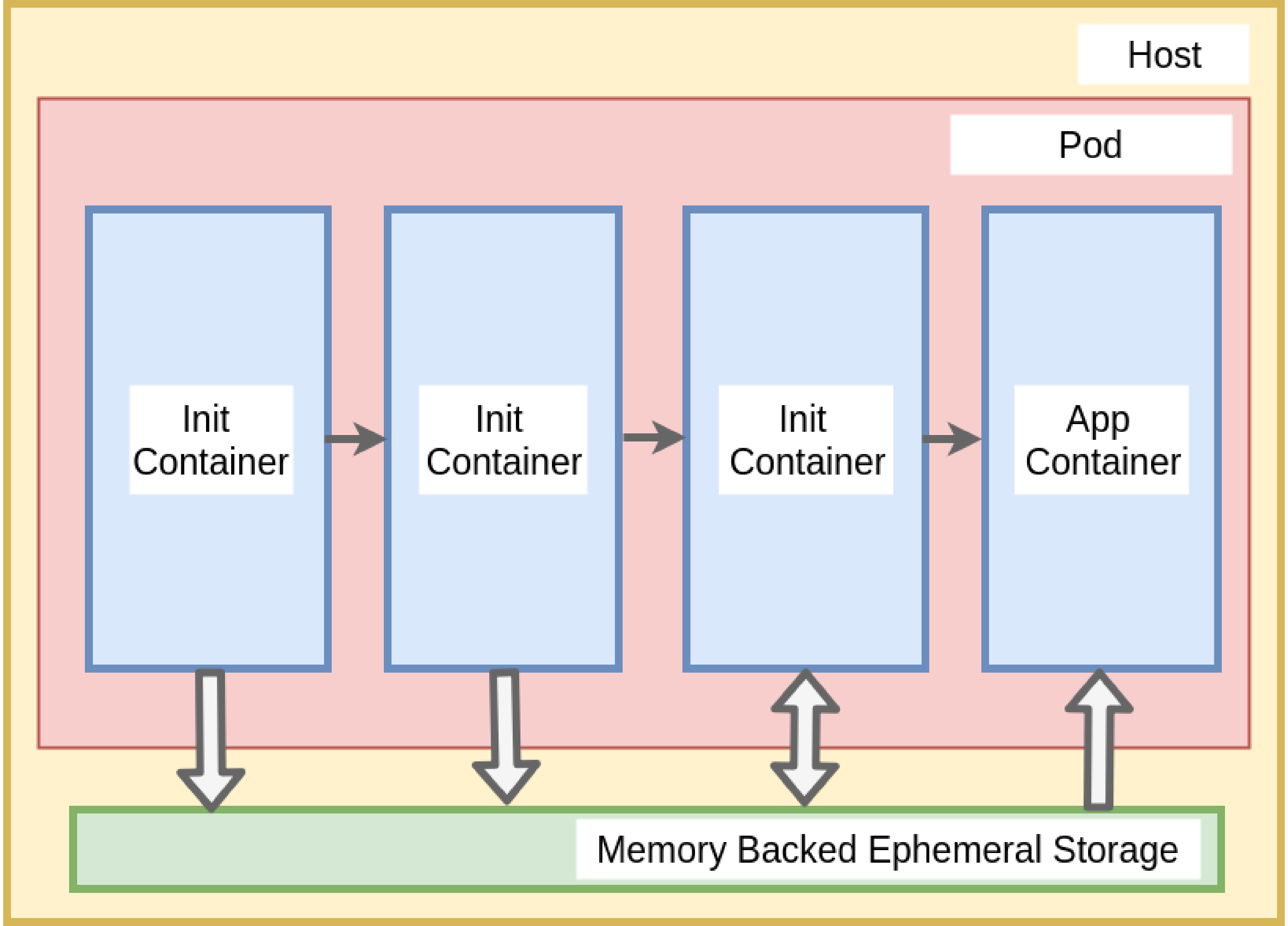
katacontainers | Kata Shim V2

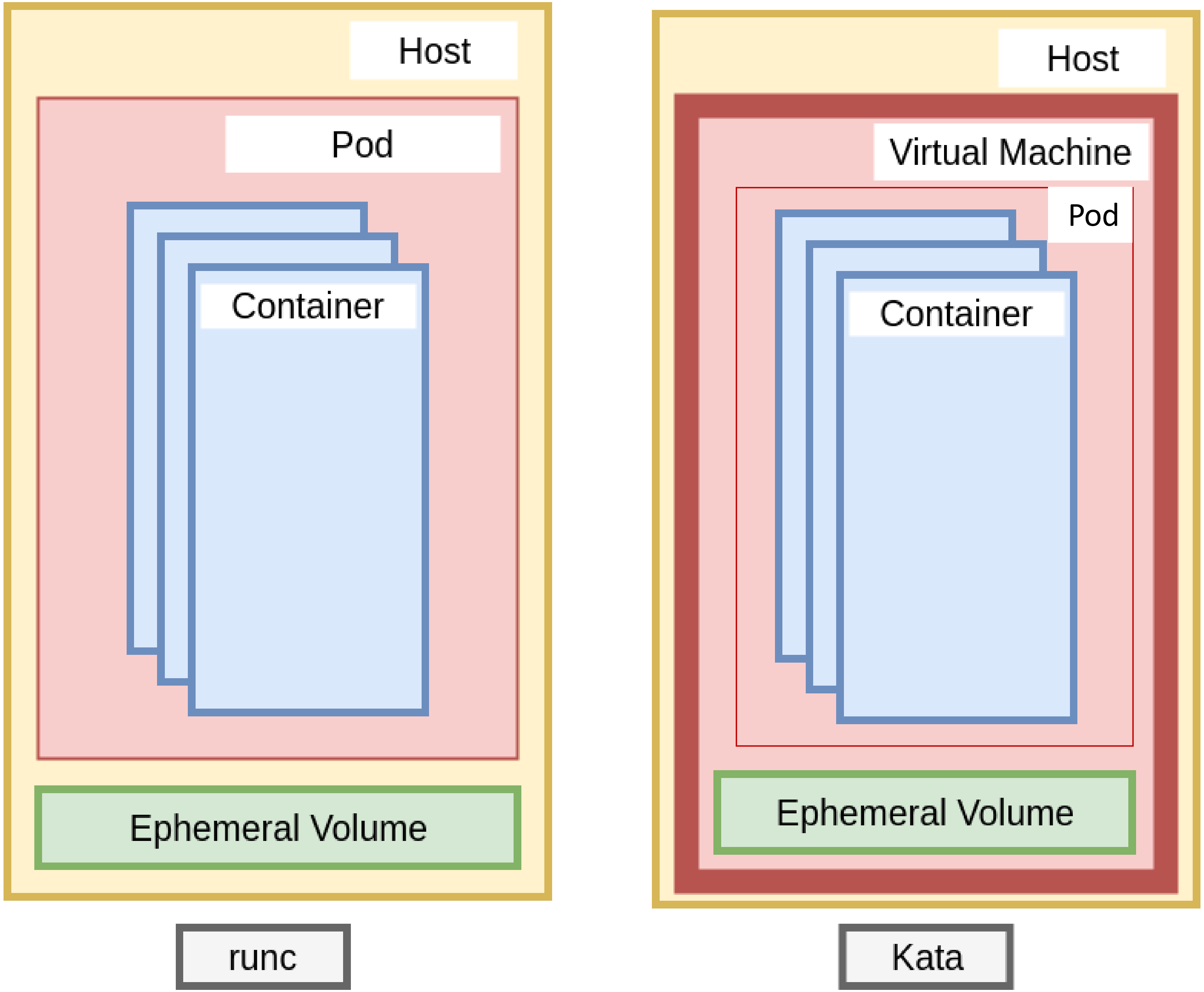


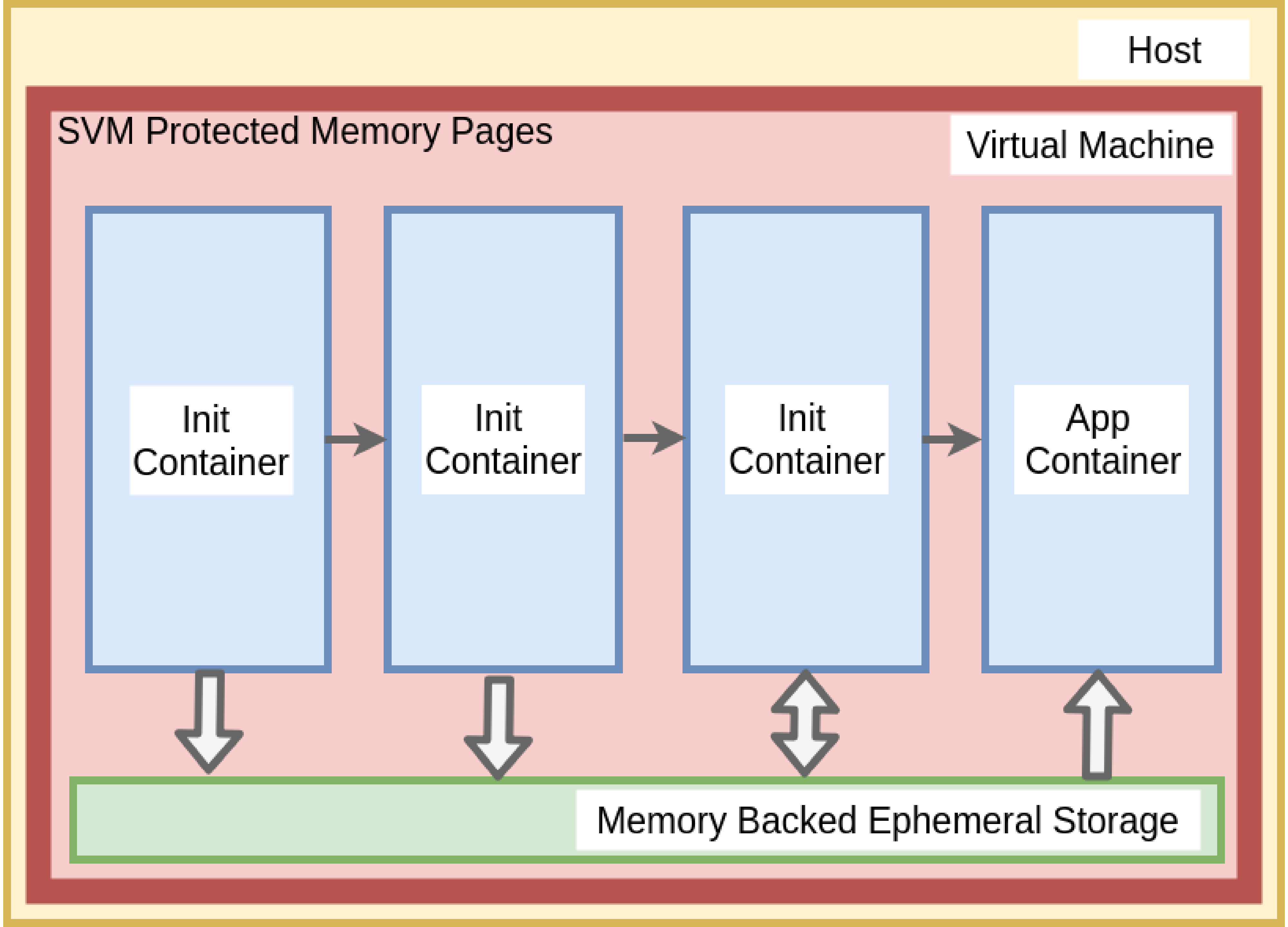
https://katacontainers.io/

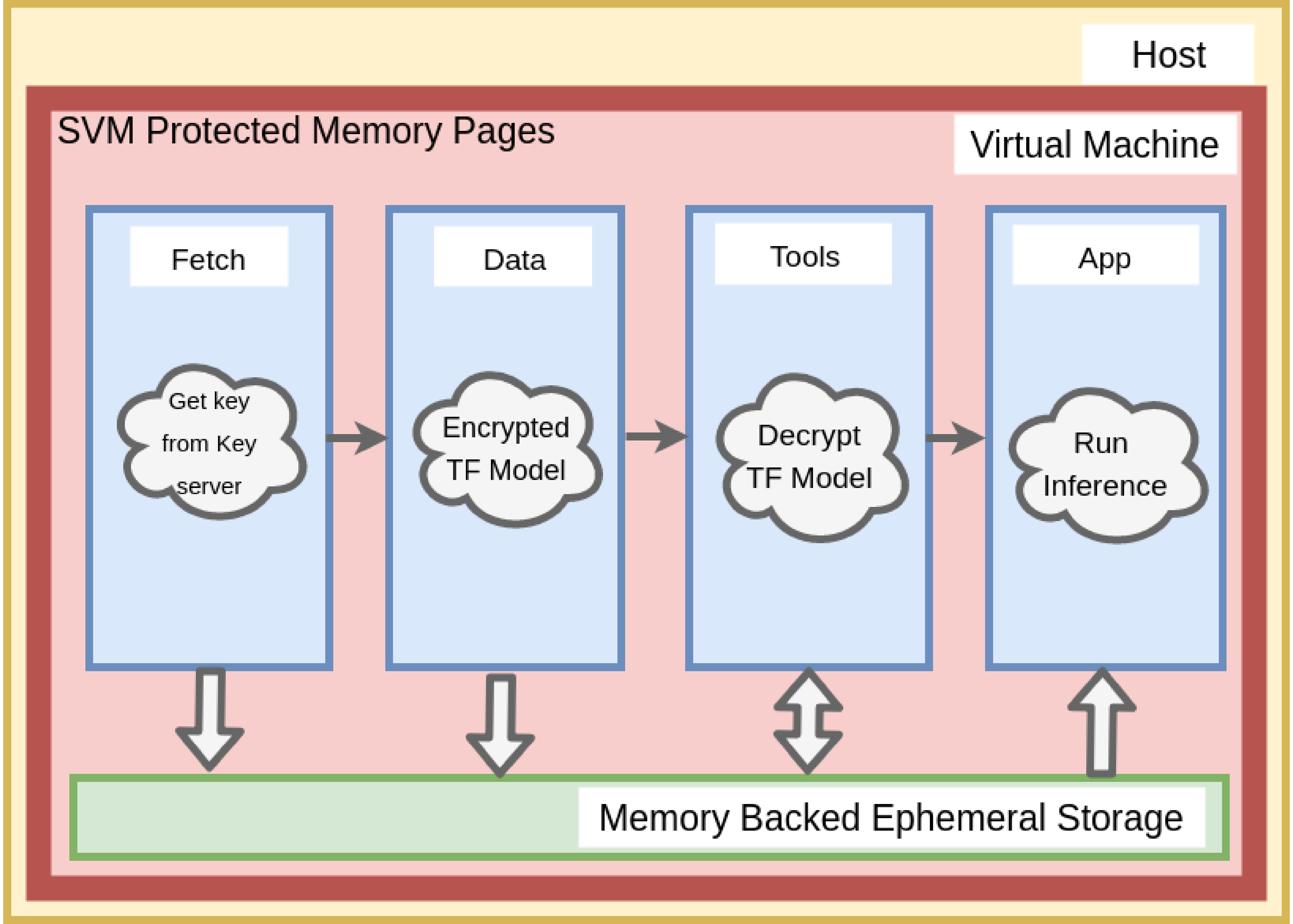
Secure Containers with Kata




Host
Pod
Init Container
Init Container
Init Container
App Container
Memory Backed Ephemeral Storage

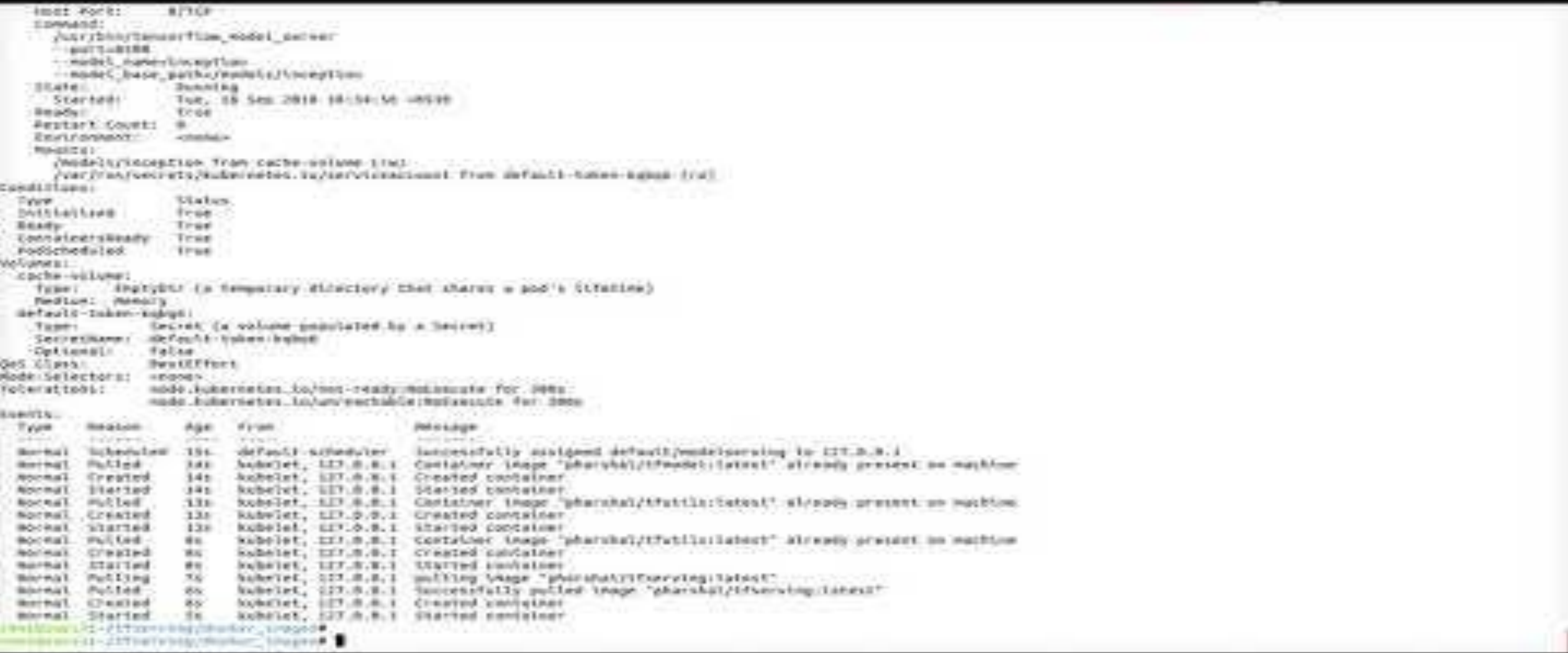

Host
Pod
Container
Ephemeral Volume
runc
Host
Virtual Machine
Pod
Container
Ephemeral Volume
Kata


Host
SVM Protected Memory Pages
Virtual Machine
Init Container
Init Container
Init Container
App Container
Memory Backed Ephemeral Storage


Host
SVM Protected Memory Pages
Virtual Machine
Fetch
Get key from Key server
Data
Encrypted TF Model
Tools
Decrypt TF Model
App
Run Inference
Memory Backed Ephemeral Storage

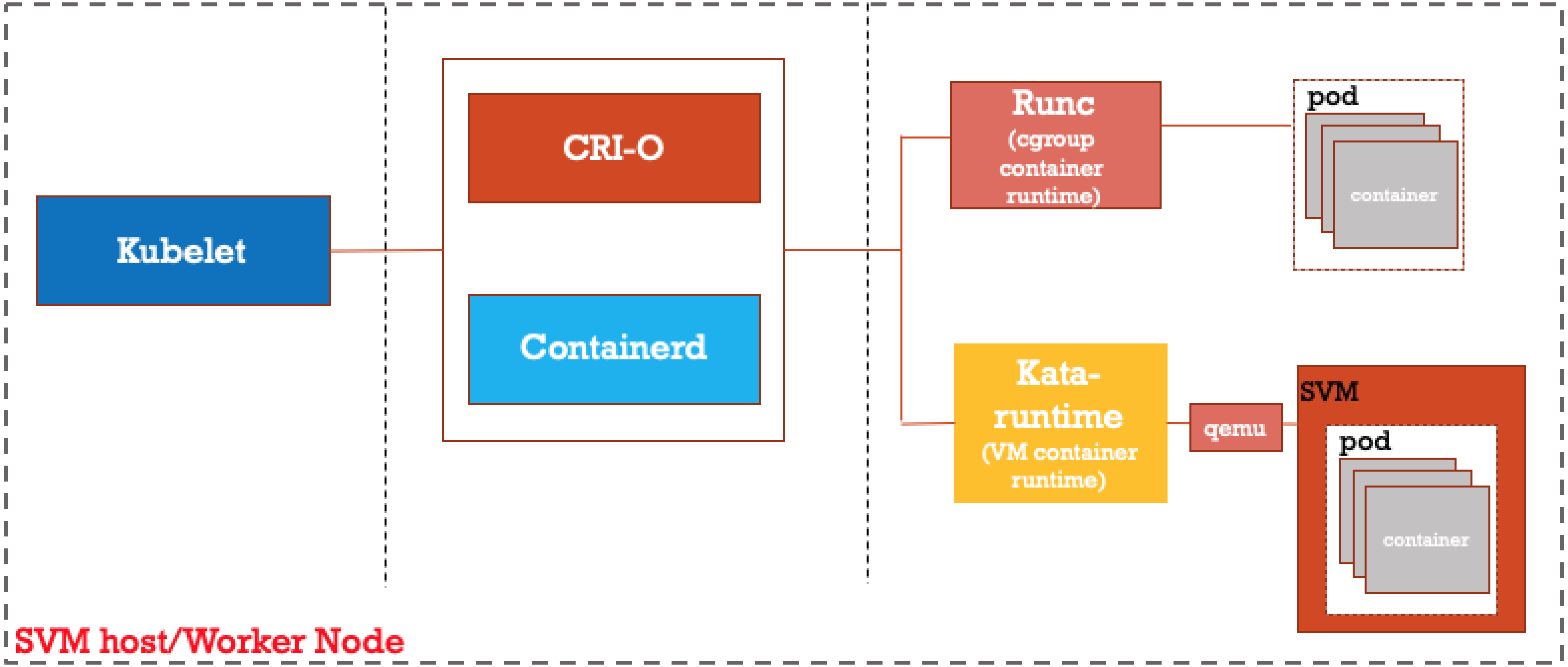
Demo

Image by Wikipedia



```
    Host Port:      0/TCP
    Command:
      /usr/bin/tensorflow_model_server
      --port=8500
      --model_name=inception
      --model_base_path=/models/inception
    State:          Running
      Started:      Tue, 18 Sep 2018 10:54:56 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /models/inception from cache-volume (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-kqbq6 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  cache-volume:
    Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:  Memory
  default-token-kqbq6:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-kqbq6
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason     Age   From                Message
  ----    ------     ----  ----                -------
  Normal  Scheduled  15s   default-scheduler   Successfully assigned default/modelserving to 127.0.0.1
  Normal  Pulled     14s   kubelet, 127.0.0.1  Container image "pharshal/tfmodel:latest" already present on machine
  Normal  Created    14s   kubelet, 127.0.0.1  Created container
  Normal  Started    14s   kubelet, 127.0.0.1  Started container
  Normal  Pulled     13s   kubelet, 127.0.0.1  Container image "pharshal/tfutils:latest" already present on machine
  Normal  Created    13s   kubelet, 127.0.0.1  Created container
  Normal  Started    13s   kubelet, 127.0.0.1  Started container
  Normal  Pulled     8s    kubelet, 127.0.0.1  Container image "pharshal/tfutils:latest" already present on machine
  Normal  Created    8s    kubelet, 127.0.0.1  Created container
  Normal  Started    8s    kubelet, 127.0.0.1  Started container
  Normal  Pulling    7s    kubelet, 127.0.0.1  pulling image "pharshal/tfserving:latest"
  Normal  Pulled     6s    kubelet, 127.0.0.1  Successfully pulled image "pharshal/tfserving:latest"
  Normal  Created    6s    kubelet, 127.0.0.1  Created container
  Normal  Started    5s    kubelet, 127.0.0.1  Started container
[illegible]:~/tfserving/docker_images#
[illegible]:~/tfserving/docker_images#
```

But...



Registry is not encrypted

Images are extracted on the host

Work in Progress

Encrypt Container images

- Ongoing effort to bring encryption to container images
- Presented in DockerCon 2019 - https://bit.ly/2LQhq3v
- KEP with Kubernetes community to add support for Encrypted Container Images
- Join us in Kubecon Shanghai 2019 where we will talk in detail

Enable the OCI runtime to pull Images

- Directly inside the confines of the SVM

References

Kata support for EmptyDir type volumes of k8s

- https://github.com/kata-containers/runtime/issues/61

Blog post

- https://mawacake.blogspot.com/2018/09/trust-tensorflow-and-cloud.html

Kubernetes KEP for Image Encryption

- https://github.com/kubernetes/community/issues/2970

IBM Power Protected Computing

- https://developer.ibm.com/articles/l-support-protected-computing/

KubeCon | CloudNativeCon

Europe 2019

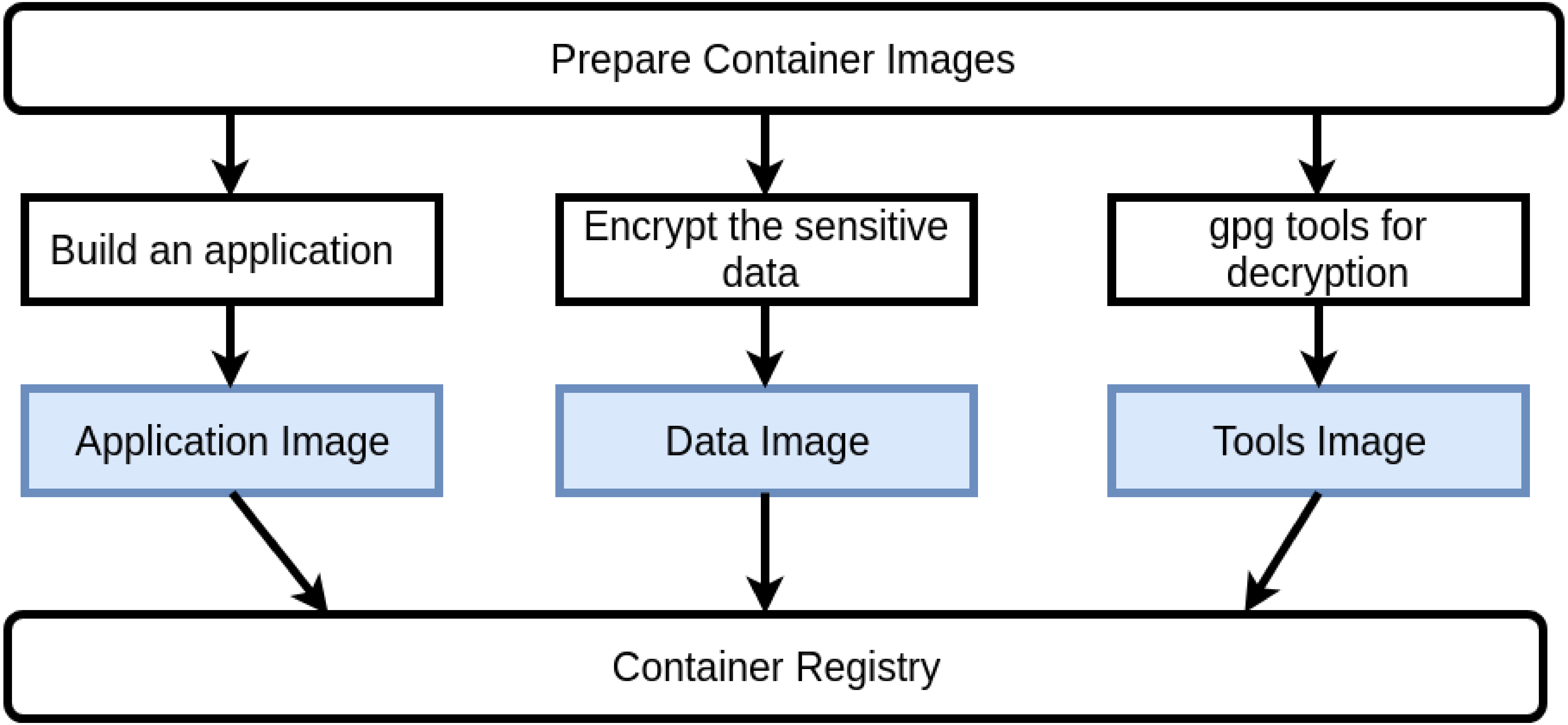
Thank You

KubeCon | CloudNativeCon

Europe 2019

Back up

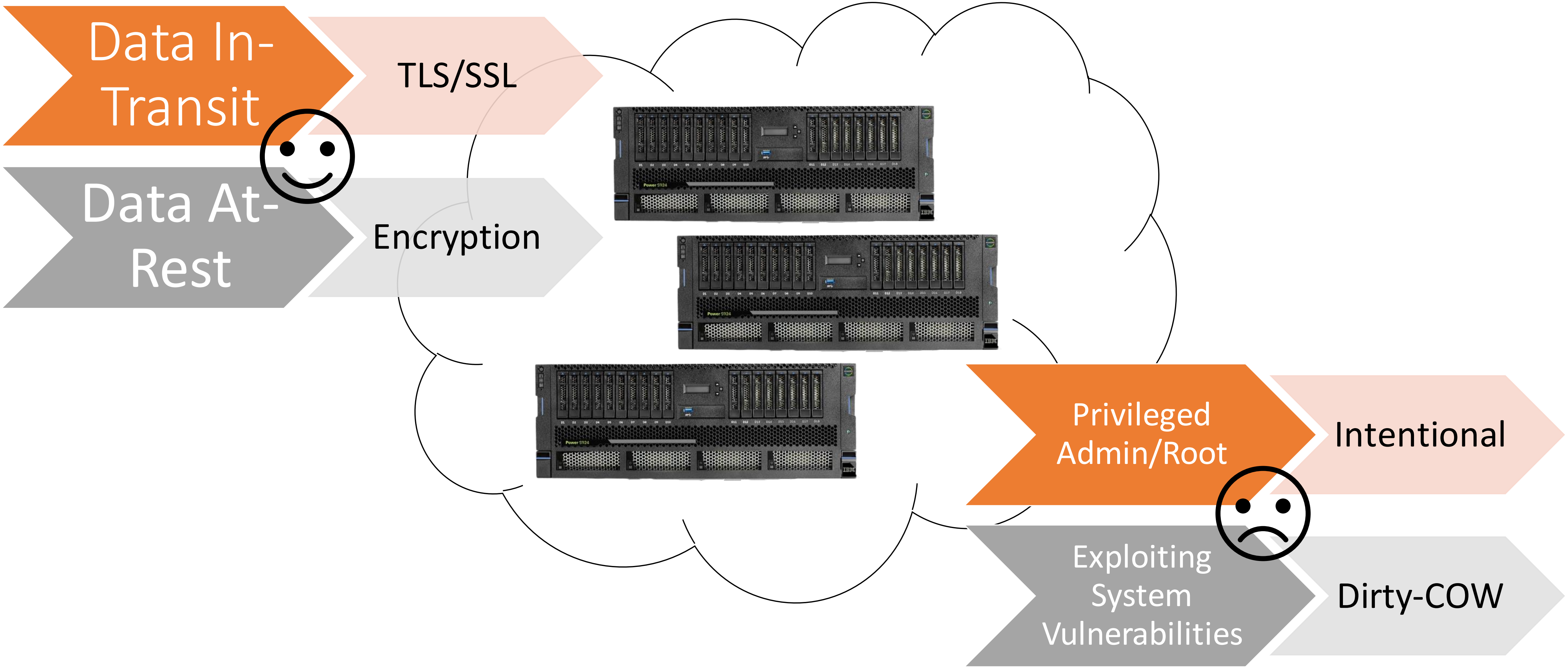
Prepare the Images



How Do We Secure Data and Code?

3F2832E97385E5AB52BB47B6B3CB7049
5615FBFADC985DBA79F9D744D0180BD5
2223D2DD724A7AF55E7867ECC4A38C65
73FD921CB12ABBA1A1D6598C958696C8
9E712DCE13216ABBD411541588A9BC67
1BBDA4568AABF582790EABD07E192288
API-KEYS
A1D70C1EB97F84155F3936F1226D4738
F657E78EF4659124CF5F35F111903AF1
FCB9728A926156B7E8F33DC938493AA3
C074663C543608262C29EB828E4FAC39
D2C57F7DF43F307A5D6E70C5A6D56013
A502C7693A78838C39B0EB7BBE2C5C21

Customer Concerns



Data In-Transit → TLS/SSL ☺

Data At-Rest → Encryption

Privileged Admin/Root → Intentional ☹

Exploiting System Vulnerabilities → Dirty-COW

If everyone stick to the rules,
just the lane marking is enough

Photo by Benny Meier on Unsplash

But when the stakes are higher,
You need a system to enforce the rules

Photo by Radio Slavonija

What do we need?

Protection of in-use data

- From other software
- Rogue administrators
- Compromised hypervisor

Untamed Root

Root User in the Cloud Systems

- Malicious root can snoop on all containers

System Vulnerabilities Can Lead to Privilege Escalation

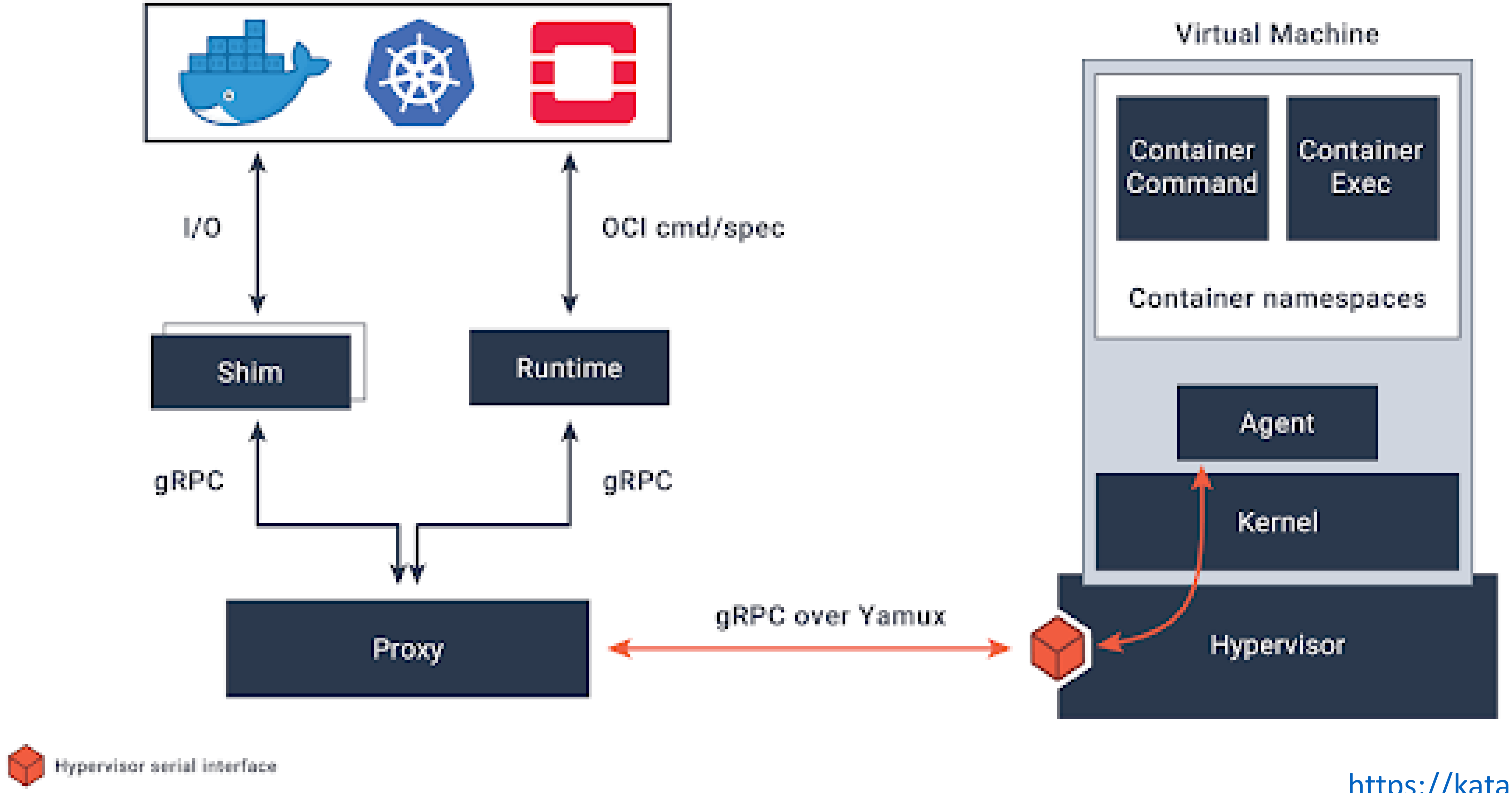
- In Multi-tenant environment this could lead to snooping on unauthorized containers
- RunC Vulnerability (CVE-2019-5736)
- Dirty COW(CVE-2016-5195)

Conflict of Interest

- What if your Cloud Provider is also your competitor

Kata

katacontainers



https://katacontainers.io/