Breaking Into Container Orchestrators

Nadeem Hussain Shaikh

SRE @Microsoft

What is Container?

- Not A Real thing, don’t confuse them with virtual machines.
- Made up of Linux primitives.
- OpenVZ, LXC, Docker, rkt, runc
- Its chroot on steroids.

Namespaces

Control what a process can see.

- PID
- Mount
- Network
- UTS
- IPC
- User
- Cgroup

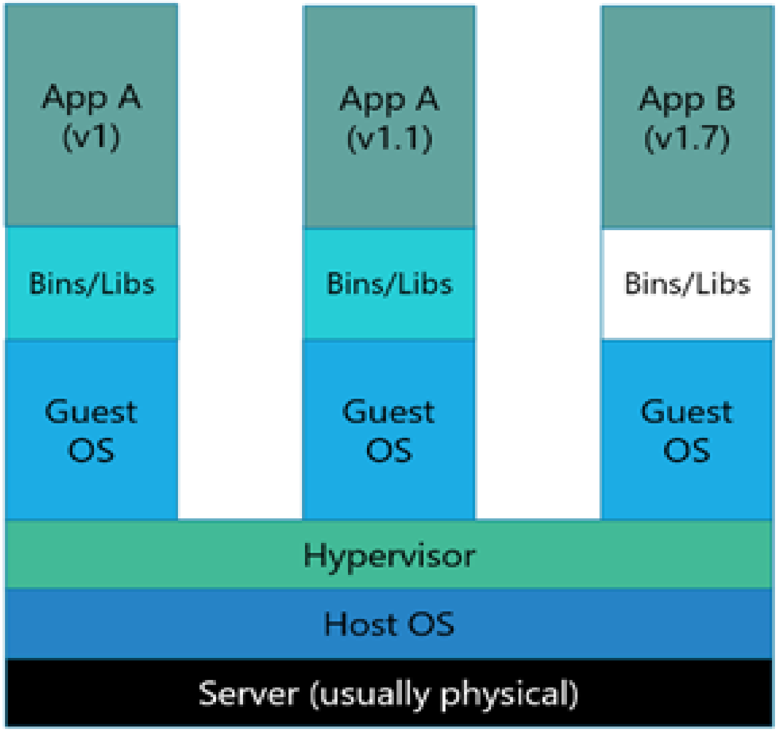
Cgroups

Control what a process can use.

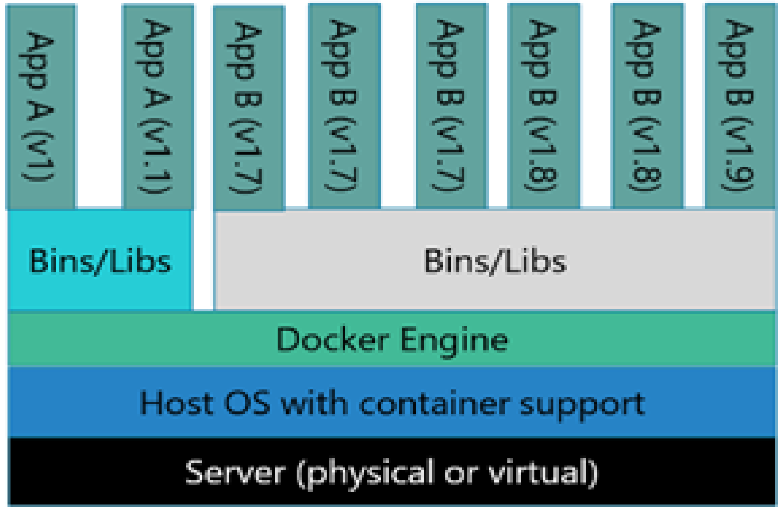
- Memory
- CPU
- Blkio
- Cpuacct
- Cpuset
- Devices
- Net_prio
- Freezer

Container vs VM

Server Virtualisation: Each app and each version of an app has dedicated OS



Containers: All containers share host OS kernel and appropriate bins/libraries



Containers are like legos

Containers come with just the pieces.

You have to build and manage with it.

Container management can be tricker

Managing it at Scale can be Nightmare

“Complexity the Worst Enemy of Security”

- Bruce Schneier

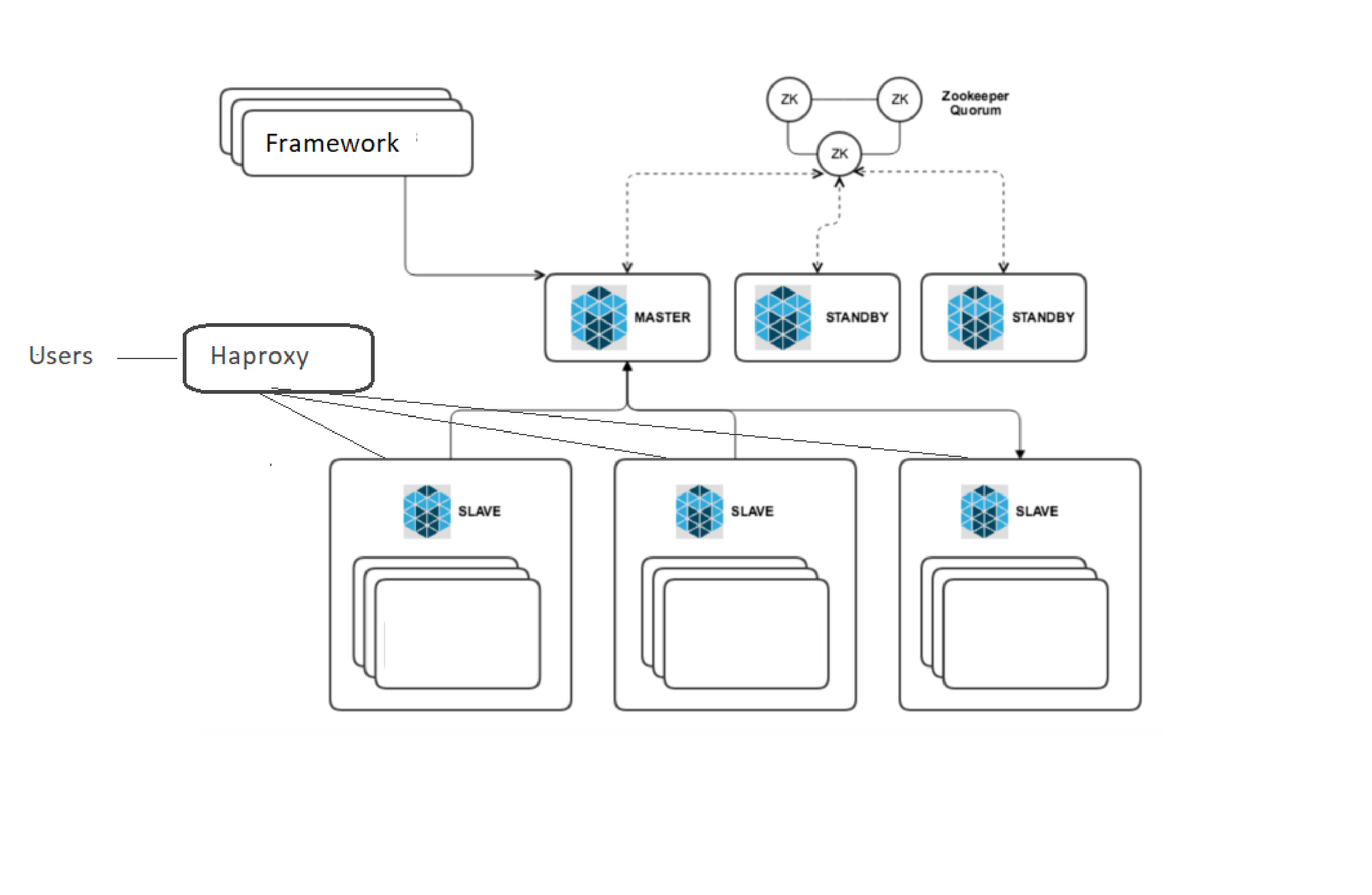
Orchestrator to Rescue

- Service Discovery
- Secret management
- Configuration management
- Logging
- Monitoring
- High-Availability
- Auto-scaling
- Stateful support
- Self-healing
- Deployment

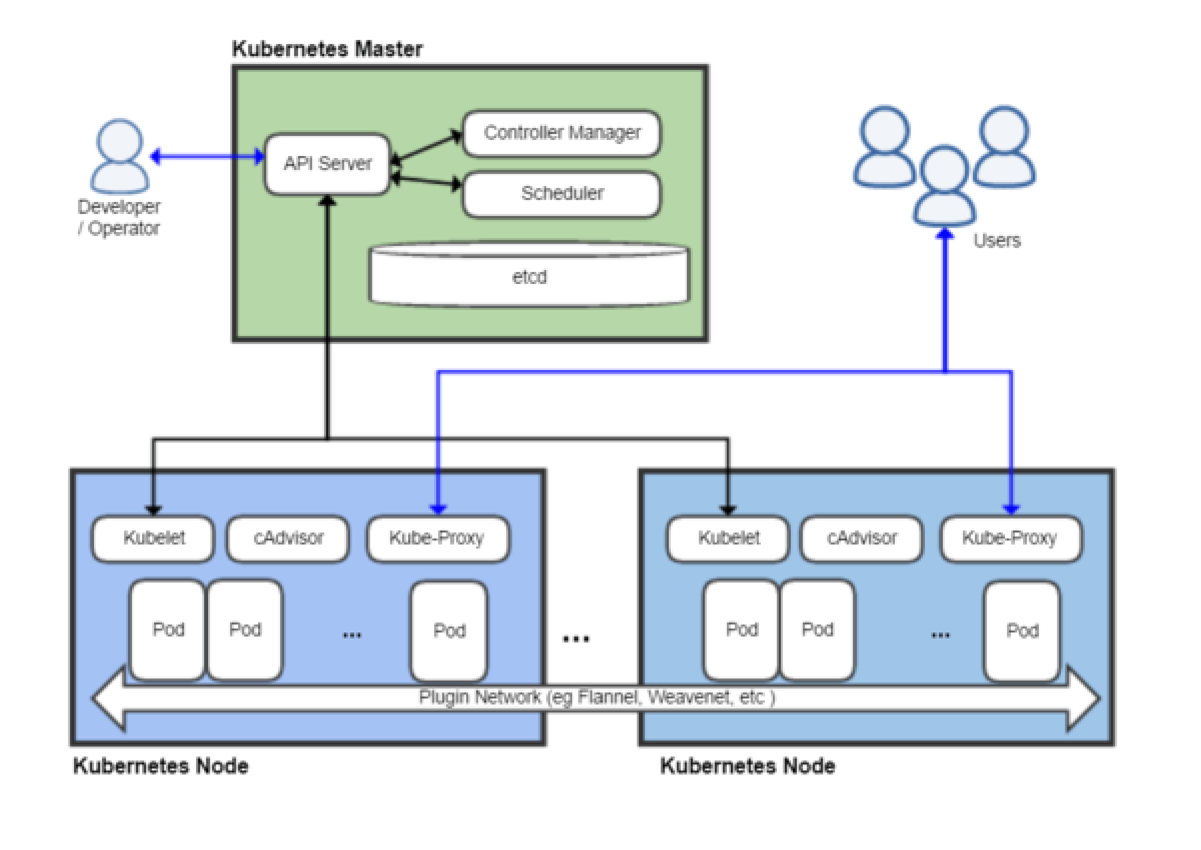
Few Orchestrator in market

- Docker Swarm
- Kubernetes
- Mesos
- Nomad
- Amazon ECS
- Azure Container Service

Mesos


Framework
Zookeeper Quorum
ZK
ZK
ZK
MASTER
STANDBY
STANDBY
Users
Haproxy
SLAVE
SLAVE
SLAVE

Kubernetes


Kubernetes Master
Developer / Operator
API Server
Controller Manager
Scheduler
etcd
Users
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
...
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Plugin Network (eg Flannel, Weavenet, etc)
Kubernetes Node
Kubernetes Node

Goals of this talk

- Awareness of possible attacks differenet orchestrators.
- Demonstrate attack on mesos and kubernetes
- Hardening method

Challenges of hardening

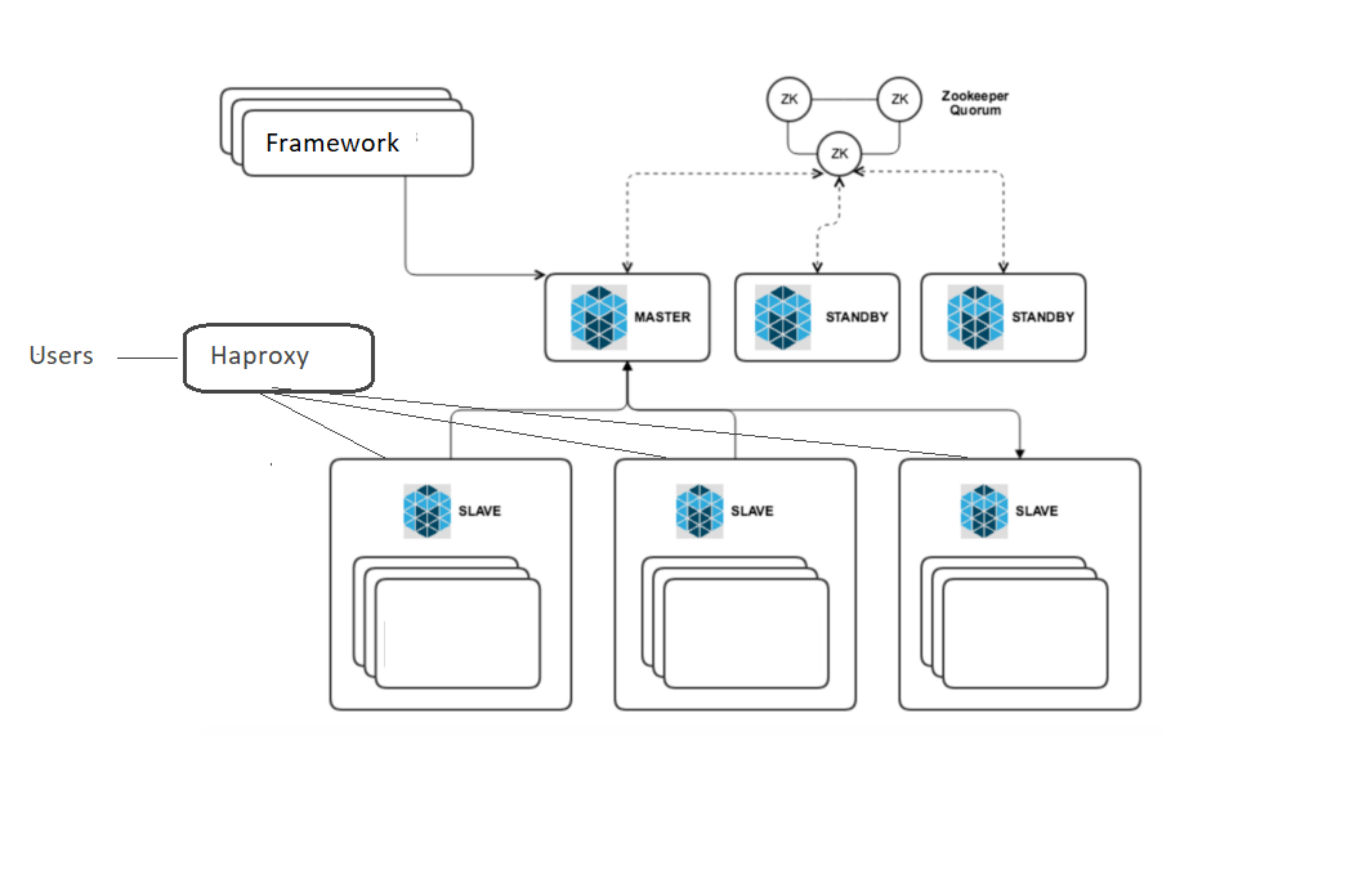
Hardening Depends upon plugins and Schdulers used , there are lots of options available

CIS benchmark for OS, Kubernetes, Docker cover core settings, but don’t consider specific implementations.

External Attacker

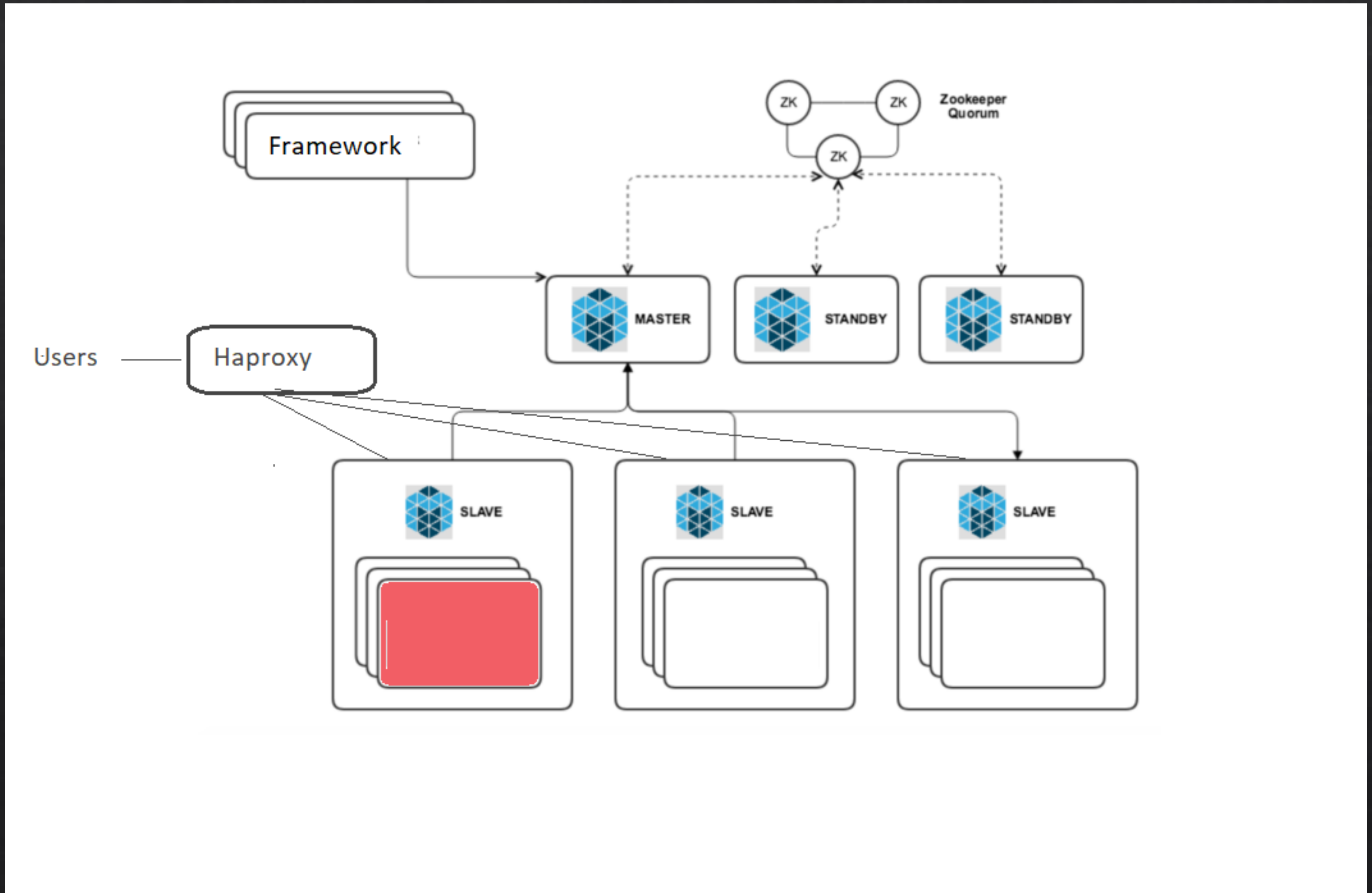
- Exploit vulnerable web app.
- Limited user ,eg. www-data.
- Enumerate cluster and check for possible methods for privilege escalation.
  - Mesos, exploit marathon API or any other framework and run custom jobs.
  - Kubernetes, exploit kubectl api with ServiceAccount.

Mesos

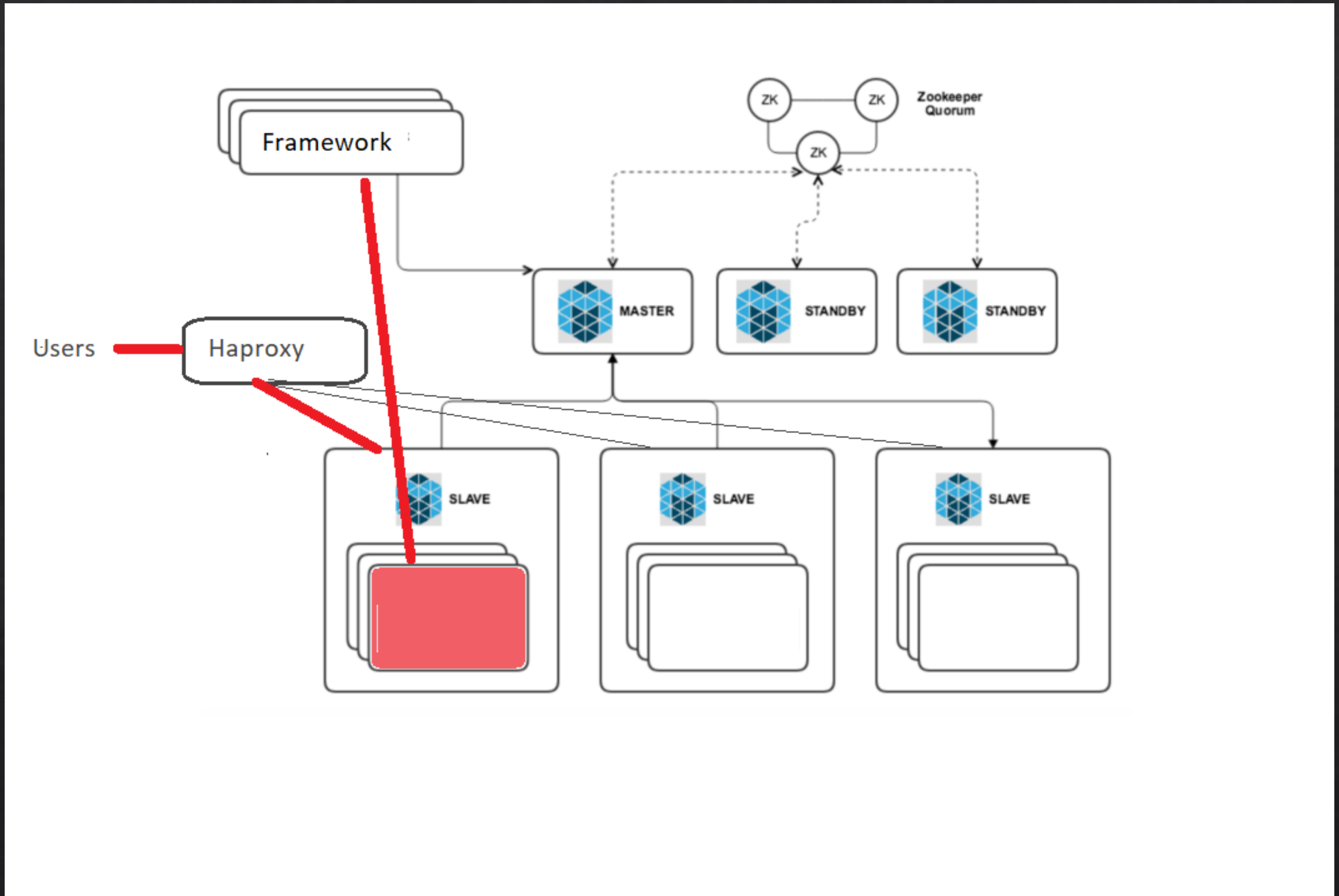


Framework
ZK
ZK
Zookeeper Quorum
ZK
MASTER
STANDBY
STANDBY
Users
Haproxy
SLAVE
SLAVE
SLAVE

Mesos


Framework
ZK
ZK
Zookeeper Quorum
ZK
MASTER
STANDBY
STANDBY
Users
Haproxy
SLAVE
SLAVE
SLAVE

Mesos

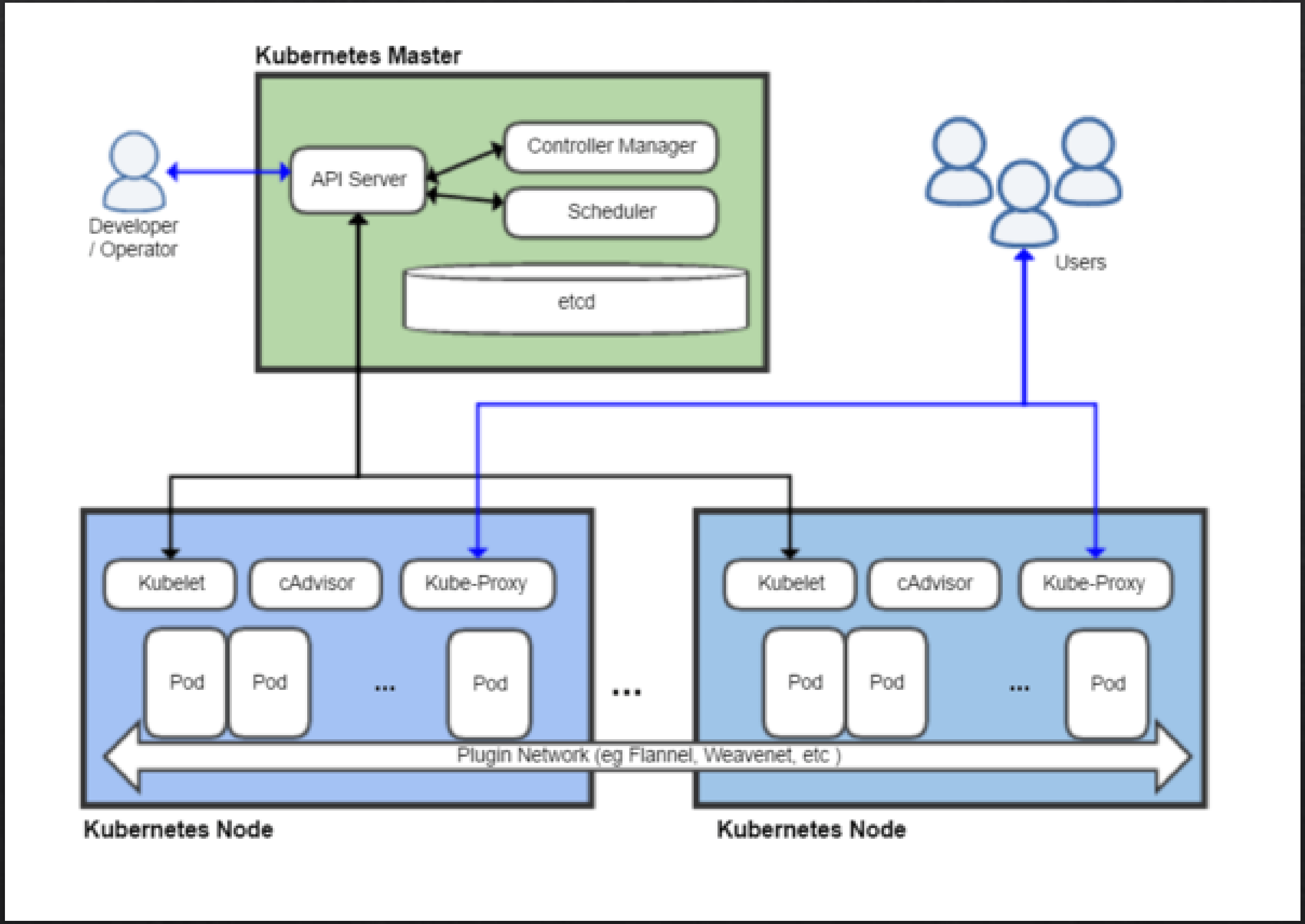


Framework
ZK
ZK
ZK
Zookeeper Quorum
MASTER
STANDBY
STANDBY
Users
Haproxy
SLAVE
SLAVE
SLAVE

Mesos

- Enable Authentication on Scheduler and mesos api.
- Secure Communication in Cluster among different Components.
- Configure security group to separate zookeeper , etcd cluster from been accessed by containers.
- Use Network Segmentation tools like Calico, Fannel .

Kubernetes


Kubernetes Master
Developer / Operator
API Server
Controller Manager
Scheduler
etcd
Users
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
...
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Plugin Network (eg Flannel, Weavenet, etc)
Kubernetes Node
Kubernetes Node

Hardening

- Enforce security settings
- Use latest stable builds
- log every thing.
- audit configuration using CIS bechmark for OS,container and Kubernetes.

Image Security

- Don't run Containers as root.
- Use private registry for image storage
- Scan Image for security vulnerabilities.
- use standard base images only

Cluster Security

- Enable RBAC policy and monitor all policy failures.
- Remove Default ServiceAccount permissions.
- Enforce TLS for all services
- Harden API server RBAC policy
- Run Etcd Cluster on dedicated nodes.
- CPU/Memory limit for every container.
- Namespaces per tenant

Network Security

- Use Network Segmentation tools like Calico, Fannel.
- ensure metadata API are not access able from containers.
- SSL offloading at Container.

thank you