



# Observabilidad en kubernetes

Fer Gleiser @fergleiser

# Acerca mio

Ing Electronico. Trabajo con Linux desde 1994. Director de Ingenieria de DCS. Me interesan la computacion, fisica y matematica

# Observabilidad en kubernetes

## Objetivos

1. Comprender la complejidad del problema
2. Aprender a usar herramientas para ayudar al diagnostico
3. Aprender que herramientas hay disponibles

# Agenda

1

Introduccion

2

Conceptos

3

Que es BPF

4

Casos de uso

5

Ejemplos

6

P y R

# Que es un container?

- AKA “Application Virtualization”
- Mecanismo para correr procesos aislado del OS del host
  - Chroot
  - namespaces
  - cgroups

# Historia

- Chroot
- BSD Jails
- Solaris zones
- Linux Containers



# Que es kubernetes

- Plataforma extensible open source para manejar contenedores
- Rollout y rollback automatico
- Self-healing
- Balanceo de carga/scalamiento horizontal

# Problemática

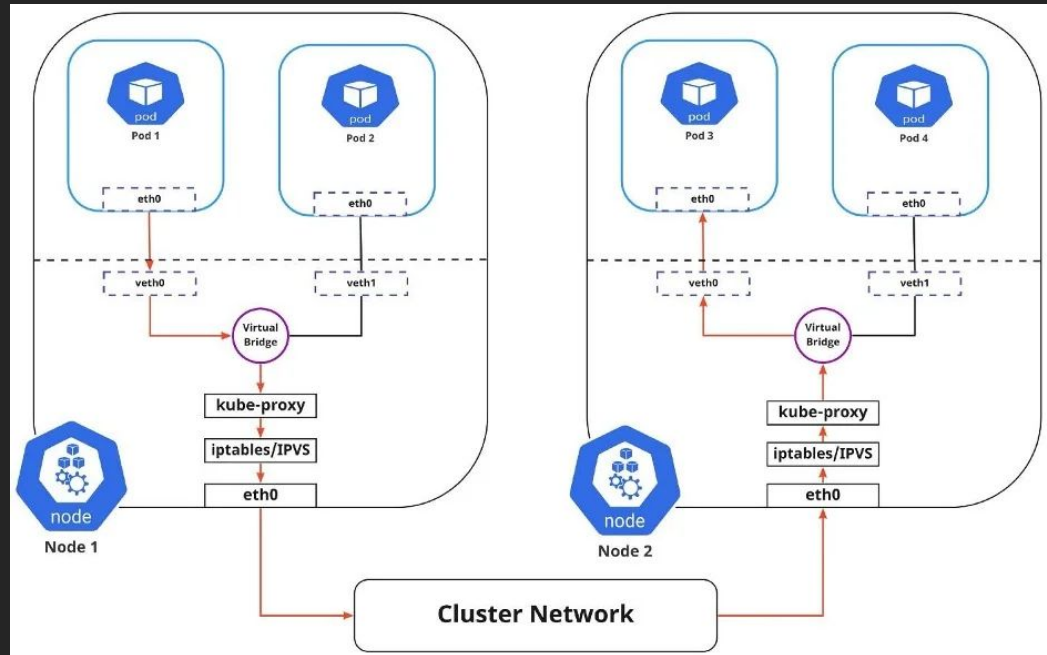
- Kubernetes es complejo
- Networking en k8s es un punto ciego
- Muchos Unknow Unknows





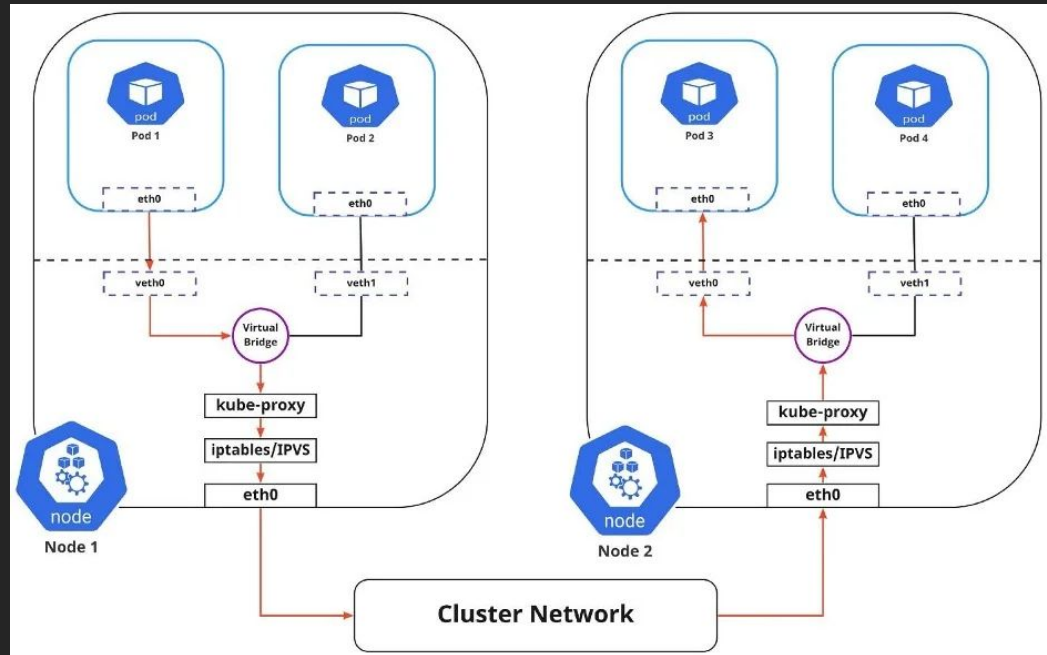
# Complejidad

Cluster to pods  
pod to pod  
service to pod  
container to container  
External to service



# Complejidad

Pods  
Nodos  
Daemonsets  
Namespaces  
Deployments





# Falta de visibilidad

- No hay observabilidad end to end
- Las herramientas tradicionales son miopes
- Tracing tradicional requiere que se sepa lo que no se sabe

# Que es la observabilidad?

Software y metodologia que permite

Loguear

Recolectar

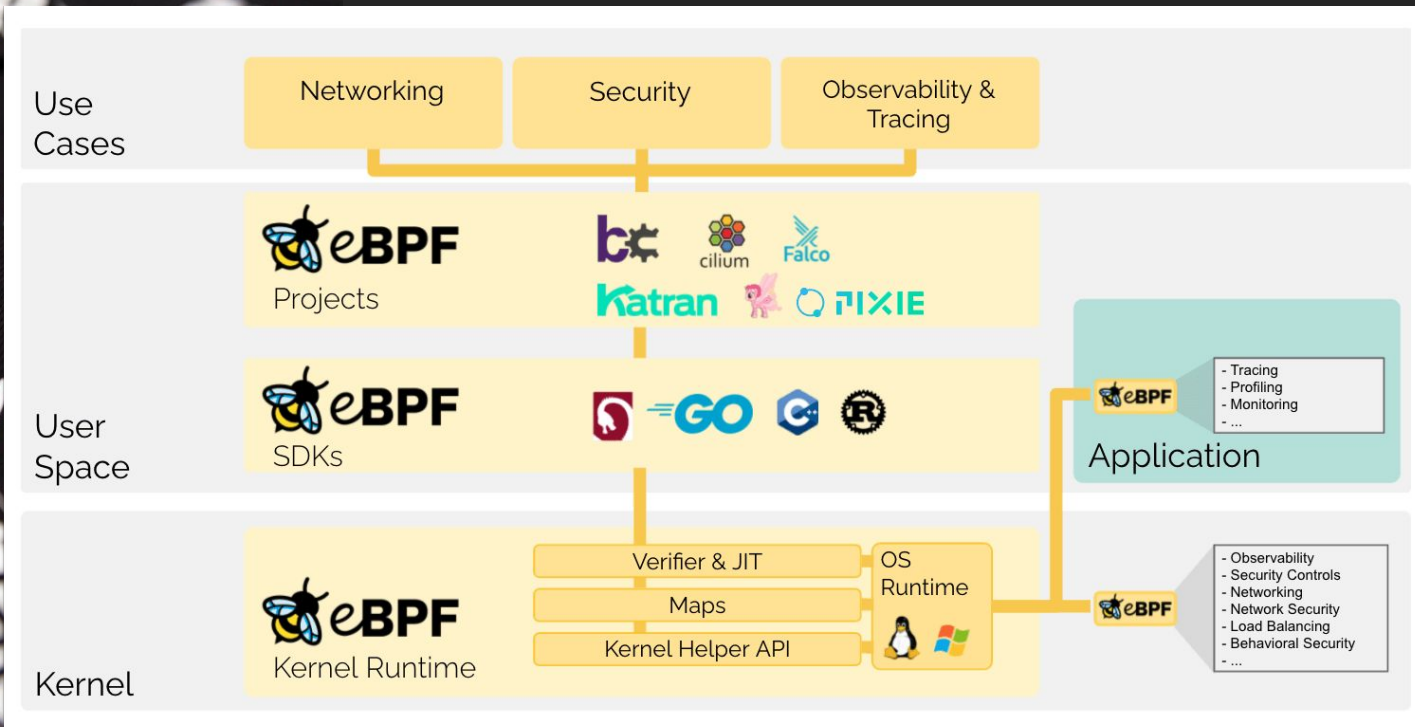
Correlacionar

analizar

# Que es eBPF?

Tecnologia para correr programas en kernel space  
Derivado de Berkeley Packet Filter  
Miles de puntos de instrumentacion

# Que es eBPF?



# Por que eBPF?

- Da visibilidad en puntos arbitrarios
- Miles de puntos de instrumentacion
- Puedo consolidar in-kernel
- Seguro para usar en produccion

# Donde instrumenta BFP

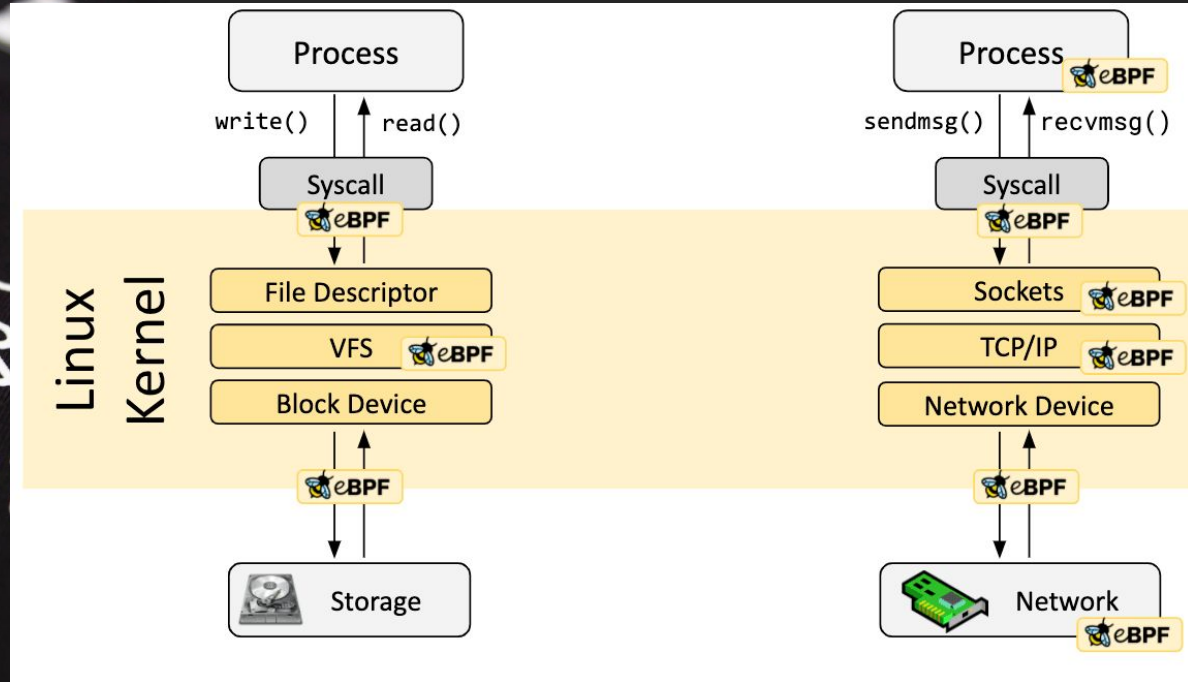
Funciones del kernel: kprobes/kretprobes

Funciones en userspace: uprobes/uretprobes

Instrumentacion estatica: tracepoints/udst



# Donde instrumenta BFP



# Casos de uso

- Performance
- Analitica
- Debugging
- Seguridad



**Pausa profilactica**



**Gracias!**