Arion异常监控系统与其他相关开源软件对比

Wei Yue

12/2022

Arion异常监控系统与其他相关开源软件的对比

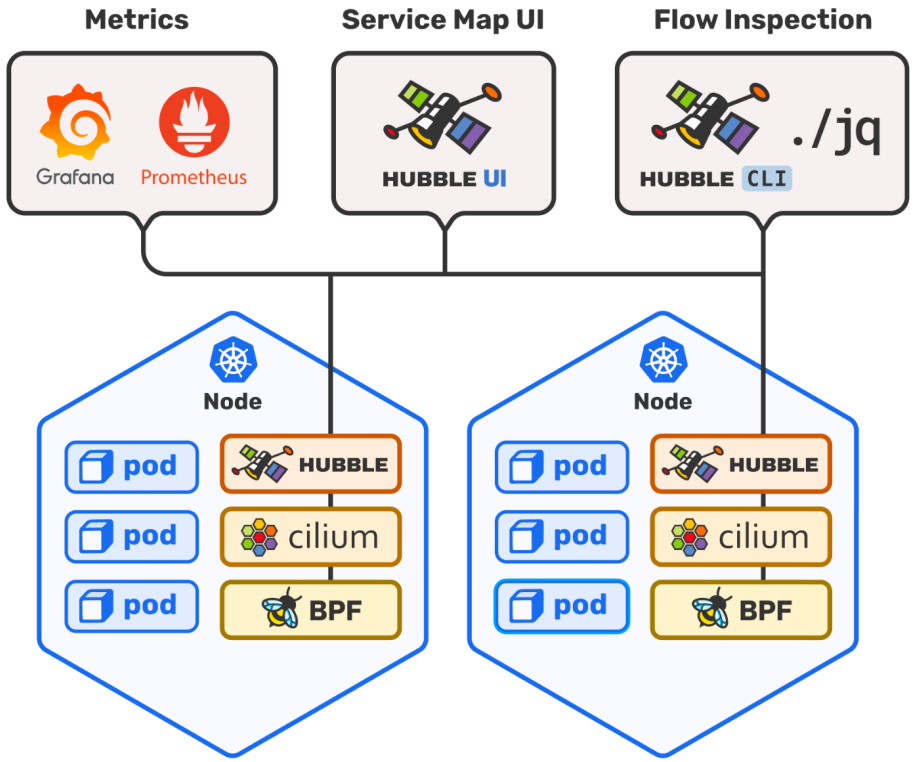
- **NetData**
  - 没有探针动态加载机制
  - 探针挂载不够灵活（例如socket只能按类挂载）
  - 不涉及resource integrity 检测与保护

- **Pixie**
  - 基于bpftrace的探针动态加载，具有一定的局限性，只能加载metrics collection的事件
  - 没有动态异常上报以及即时处理的能力
  - 不涉及resource integrity 检测与保护

- **Arion Monitoring**
  - 灵活的动态创建与部署监控event，监控事件不受限
  - 对异常检测有即时上报以及本地处理的能力
  - advanced resource access control机制（例如对eBPF map的保护）

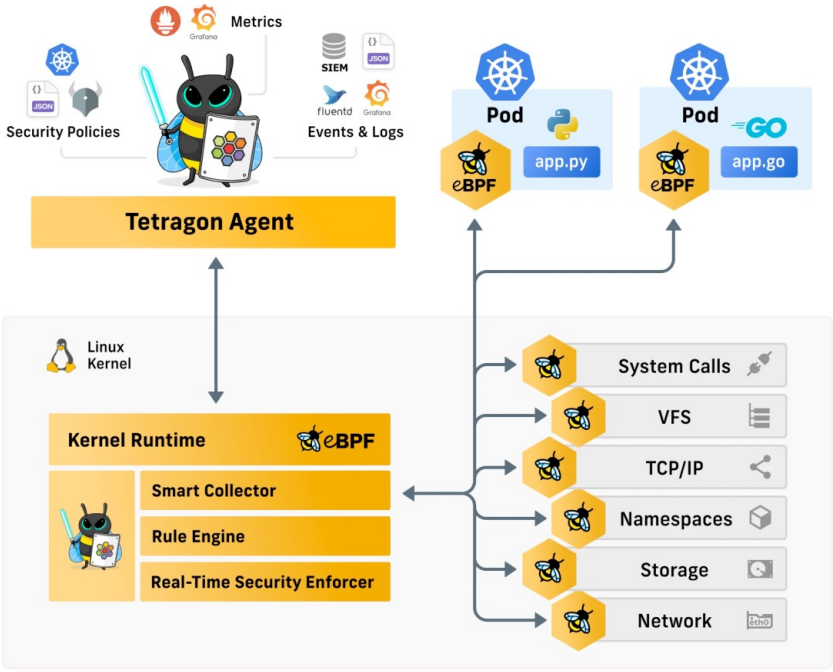
Arion异常监控系统与其他相关开源软件的对比

- **Hubble**
  - Cilium的eBPF based网络监控平台
  - 主要监控能力包括：
    - service dependency graph
    - Metrics & Monitoring
    - Flow visibility
  - 不具有动态创建以及部署监控event的能力
  - 不涉及resource integrity 检测与保护

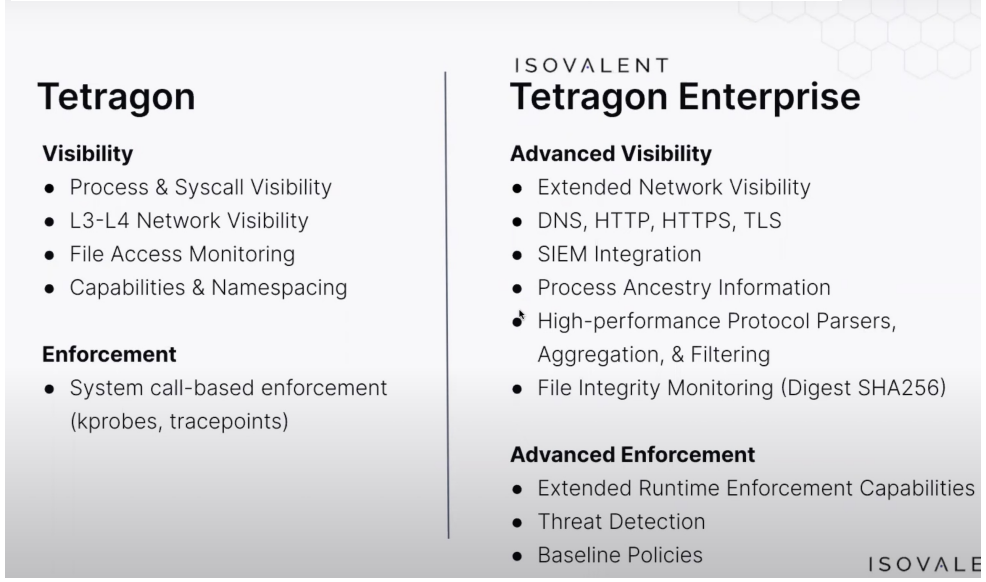


Arion异常监控系统与其他相关开源软件的对比

- **Tetragon**
  - Cilium 针对Hubble的局限性，添加了实时性监控以及Runtime enforcement
  - 有Enterprise和开源版，开源版功能 is limited
  - 有了对敏感文件访问监控的能力（需集成Splunk）
  - 理念上跟Arion monitoring system最接近（Tetragon在我们的proposal后开源），我们在propose后的技术探讨方案中所解决的一些问题也是Tetragon在试图解决的问题

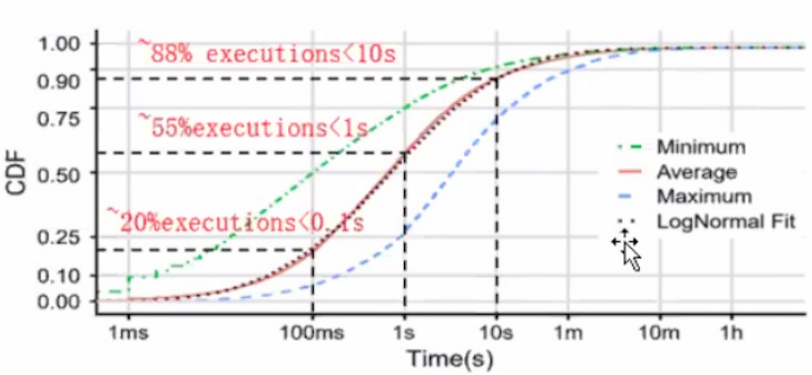


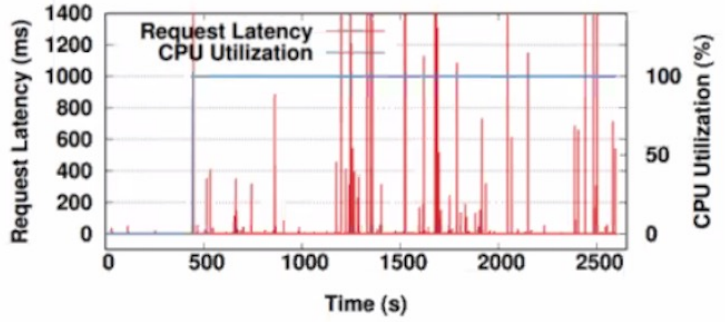
- *Tetragon tightly coupled with Cilium*
- *Tetragon提供通用solution*
- *Arion 在场景deep dive基础上的 optimal solution*
- *Our advantage*
  - *快速闭环反馈机制基础上的动态部署*
  - *Advanced resource control mechanism*



Arion异常监控系统对serverless平台监控的必要性

- Serverless平台特性对监控的要求
  - 需要细颗粒的监控（大量Function的生命周期短）
    - *88% < 10s*
  - 功能链复杂，平台全量监控代价太大，需要有<秒级，甚至毫秒级的动态调整监控目标的能力





- 传统监控方案面对Serverless平台的局限性
  - 采样率不满足细颗粒要求
    - 普遍以分钟级监控容器；
    - 提升采样率会导致：
      - 监控CPU占用率高
      - 数据量爆炸
      - 大量“无用”数据，资源浪费
  - 不具备闭环动态调整监控目标的能力，对Serverless场景下异常事件的监控能力不强。

- Arion监控方案专门针对Serverless平台监控需求
  - 快速闭环反馈机制基础上的动态部署提供快速动态调整监控目标的能力
  - 利用eBPF技术大幅度降低监控资源占用率
  - Advanced resource control mechanism提供进一步的安全监控能力
  - 结合传统监控平台一起解决serverless监控需求，是补全，不是替换。

Arion异常监控系统potential examples

- 基于eBPF的监控系统在降低监控开销的同时极大的增强了传统监控系统的监控能力;

- Serverless场景对监控需求更高，Arion的异常监控系统的主动闭环机制寻求更进一步适配serverless场景的特性，实现在大部分function的生命周期内进行主动反馈，尽量避免damage的落实或扩大;

- 例如：我们的监控机制对第三方用户容器功能可以提供一系列预先定制的安全监控和运行检验，一旦发现不正常情况(untheorized access, wrong path, CPU surge, process crash, etc.), 主动上报control agent，control agent根据定制的逻辑，做出相应处理， 例如：在上游function chain添加相应eBPF 处理功能，对当前function launching添加更细颗粒监控等等。