金山办公基于 DeepFlow 的可观测性实践

肖圆 金山办公高级研发 私有化可观测平台负责人

CONTENTS

01 可观测建设现状

- 架构调整：从k8s微服务架构逐渐演变为单元化架构
- 战略调整：可观测作为核心能力模块成为公司级项目

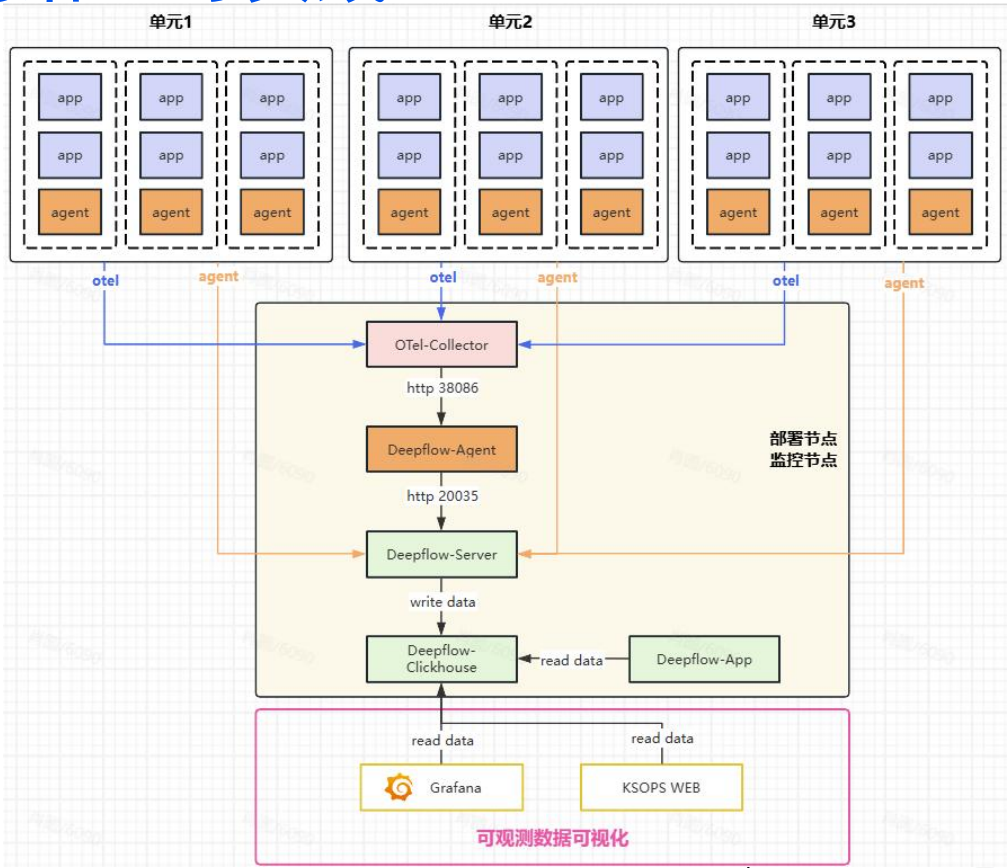
为配合架构演进需求，技术团队将基于 DeepFlow 可观测平台，在纯Docker环境（采用host网络模式）中构建新一代的全栈可观测体系，为架构转型提供坚实的技术保障。

CONTENTS

02 可观测落地与实践

部署架构(1)

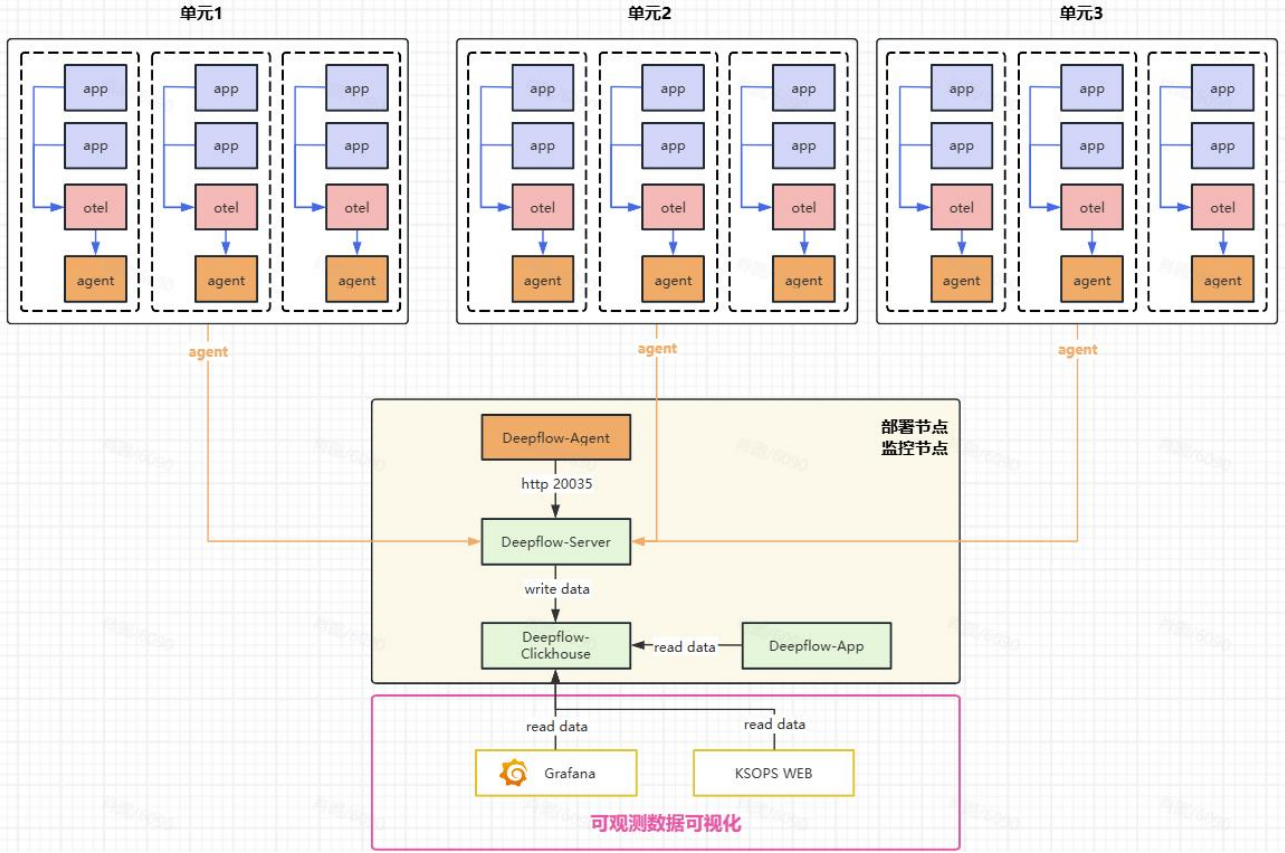
多otel sdk
|
单个collector
|
单个agent
|
单个server
|
单个ck



02 可观测落地与实践

部署架构(2)

单个otel sdk
|
单个collector
|
单个agent
|
单个server
|
单个ck



02 可观测落地与实践

部署方式

server-docker-compose.yaml:

```
version: '3.8'
services:
  clickhouse:
    image: 'deepflow-clickhouse:v23.8.7.24'
    network_mode: host
    .......
    deploy:
      resources:
        limits:
          cpus: '4'
          memory: '32g'
  deepflow-server:
    image: 'reg.wps.lan:5000/wps/deepflow-server:v6.6'
    network_mode: host
    user: "65534"
    environment:
      DEEPFLOW_SERVER_RUNNING_MODE: STANDALONE
      K8S_NODE_IP_FOR_DEEPFLOW: 'x.x.x.x'
      K8S_POD_IP_FOR_DEEPFLOW: '::1'
      K8S_NAMESPACE_FOR_DEEPFLOW: deepflow
      K8S_NODE_NAME_FOR_DEEPFLOW: deepflow-host
      K8S_POD_NAME_FOR_DEEPFLOW: deepflow-server
    ......
```

```
    deploy:
      resources:
        limits:
          cpus: '4'
          memory: '16g'

  deepflow-app:
    image: 'deepflow-app:v6.6'
    network_mode: host
    ......
    deploy:
      resources:
        limits:
          cpus: '2'
          memory: '4g'

  otel-collector:
    image: 'otel-collector:v0.101.0'
    network_mode: host
    .....
    deploy:
      resources:
        limits:
          cpus: '4'
          memory: '4g'
```

agent-docker-compose.yaml:

```
version: '3.2'
services:
  deepflow-agent:
    image: 'deepflow-agent:v6.6'
    privileged: true
    network_mode: host
    pid: host
    cap_add:
      - SYS_ADMIN
      - SYS_RESOURCE
      - SYS_PTRACE
      - NET_ADMIN
      - NET_RAW
      - IPC_LOCK
      - SYSLOG
    volumes:
      - "/data/kube/deepflow-agent/deepflow-
agent.yaml:/etc/deepflow-agent/deepflow-agent.yaml:ro"
      - /sys/kernel/debug:/sys/kernel/debug:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
```

02 可观测落地与实践

部署配置

agent-group-config.yaml:

```
inputs:
  resources:
    workload_resource_sync_enabled: true
  proc:
    enabled: true
    process_matcher:
    - enabled_features:
      - ebpf.profile.on_cpu
      match_regex: ^(deepflow|docmini|mysqld|docs|node|nginx|docs-file-
cps|editserver|transcoder|cloudprovider|htmlserver|apiserver)
      only_in_container: false
  integration:
    enabled: true
    listen_port: 38086
global:
  limits:
    max_memory: 2048
    max_millicpus: 2000
  circuit_breakers:
    relative_sys_load:
      trigger_threshold: 2.0
      recovery_threshold: 1.5
      metric: load15
```

```
communication:
    proxy_controller_port: 20035
    ingester_port: 20033
    proxy_controller_ip: "x.x.x.x"
    ingester_ip: "x.x.x.x"
processors:
  request_log:
    tag_extraction:
      tracing_tag:
        x_request_id:
        - X-Request-Id
      http_endpoint:
        match_rules:
        - keep_segments: 3
          url_prefix: ''
      obfuscate_protocols: [MySQL,Redis,Kafka]
    application_protocol_inference:
      enabled_protocols: [MySQL,Redis,Kafka,HTTP,HTTP2,gRPC]
```

02 可观测落地与实践

1、Docker 模式下无法采集容器资源信息

说明：

1. 当前 v6.6 版本中暂未加入对 Docker 环境的适配，无法识别 Docker 容器的名称。
2. 目前对于 Docker 容器的识别可通过 eBPF 采集到的进程名称（容器需使用宿主机网络）+ IP + PORT

Request List

Flame Graph	Status	Response Exceptior	Response Delay	Start time ↓	Client ip	Client Process	Server ip	Server Process	server_port	Data type	Enum(observation_p
id-75193...	Success	--	26 µs	2025-06-24 10:36:27.997315	10. 3	--	10. 3	nginx	22307	eBPF	Server Process
id-75193...	Client Error	Not Found	63 µs	2025-06-24 10:36:27.997207	10. 3	--	10. 3	cal om-w	22104	eBPF	Server Process
id-75193...	Success	--	84 µs	2025-06-24 10:36:27.997204	10. 3	--	10. 3	nginx	22155	eBPF	Server Process
id-75193...	Client Error	Not Found	395 µs	2025-06-24 10:36:27.997019	10. 3	keepalived	10. 3	--	22104	eBPF	Client Process
id-75193...	Client Error	Not Found	87 µs	2025-06-24 10:36:27.996950	10. 3	--	10. 3	editproxy	20024	eBPF	Server Process
id-75193...	Success	--	486 µs	2025-06-24 10:36:27.996937	10. 3	keepalived	10. 3	--	221	eBPF	Client Process
id-75193...	Client Error	Not Found	174 µs	2025-06-24 10:36:27.996905	10. 3	keepalived	10. 3	--	200	eBPF	Client Process
id-75193...	Success	--	627 µs	2025-06-24 10:36:27.996855	10. 3	keepalived	10. 3	--	22307	eBPF	Client Process
id-75193...	Success	--	50 µs	2025-06-24 10:36:27.996850	10. 3	--	10. 3	ks apd	22270	eBPF	Server Process
id-75193...	Success	--	234 µs	2025-06-24 10:36:27.996809	10. 3	keepalived	10. 3	--	22270	eBPF	Client Process

宿主 IP　容器进程　容器端口

02 可观测落地与实践

2、IPV6 环境部署适配性问题

说明：部署后发现 Server 端没有 controller ip，经排查后发现 Server 配置文件中存在多处配置格式不统一，不太易用。与社区及时沟通后将配置格式统一为后解决问题。

```
alse, kubernetes_watch_policy: 0, group_id: g-c46134c14b) not found in cache. NAME:deepflow-agent-ce
750146118
2025-06-20 17:21:24.208 [INFO] [trisolaris.vtap] vtap.go:1368 start vtap register
2025-06-20 17:21:24.208 [ERRO] [trisolaris.vtap] vtap.go:223 config NodeIP is null
2025-06-20 17:21:24.208 [ERRO] [trisolaris.vtap] vtap.go:1343 controller not found region
2025-06-20 17:21:24.208 [INFO] [trisolaris.vtap] vtap.go:1375 end vtap register
2025-06-20 17:21:24.234 [INFO] [tagrecorder] dictionary.go:81 tagrecorder update ch dictionary
```

改之前配置

```
controller:
  election-name: deepflow-server
  election-namespace: deepflow
  grpc-max-message-length: 104857600
  grpc-port: 20035
  kubeconfig:
  listen-port: 20417
  clickhouse:
    database: flow_tag
    host: "[2408:8656:22df:ff01::14:fe06]"
    port: 19000
    user-name: default
    user-password:
  mysql:
    auto_increment_increment: 1
    database: deepflow
    host: "2408:8656:22df:ff01::14:fe06"
    port: 4306
    timeout: 30
    user-name: root
    user-password:
```

改之后配置

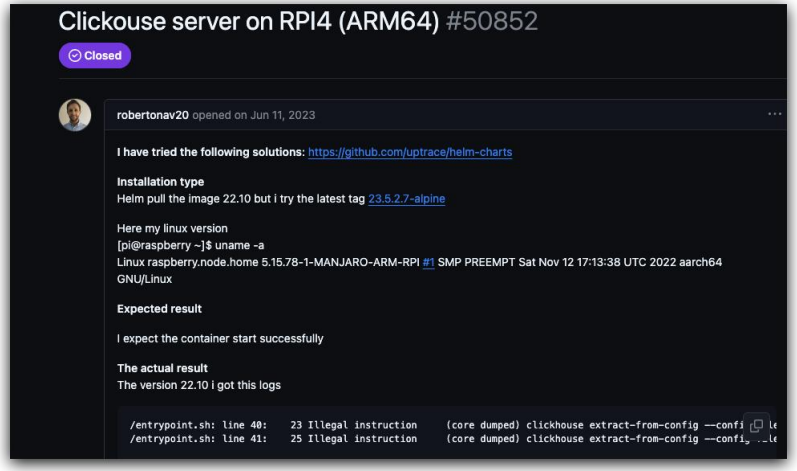
```
controller:
  election-name: deepflow-server
  election-namespace: deepflow
  grpc-max-message-length: 104857600
  grpc-port: 20035
  kubeconfig:
  listen-port: 20417
  clickhouse:
    database: flow_tag
    host: "2408:8656:22df:ff01::14:fe06"
    port: 19000
    user-name: default
    user-password:
  mysql:
    auto_increment_increment: 1
    database: deepflow
    host: "2408:8656:22df:ff01::14:fe06"
    port: 4306
    timeout: 30
    user-name: root
    user-password:
```

02 可观测落地与实践

3、 ARM 环境部署时，ClickHouse 需要单独拉 ARM 镜像

说明：DeepFlow 自身相关的组件镜像均支持 ARM 环境，其他中间件 MySQL 与 CK 需要从对应社区同步镜像。DeepFlow 默认提供的是 X86 镜像，解决方式分为两种：

1. 去对应社区拉取镜像：例如可去鲲鹏官网查询同步的 ARM 镜像版本
2. 使用自建 CK 环境或自定义构建 ARM 版本镜像。自定义镜像可参考以下方式：
    - https://github.com/ClickHouse/ClickHouse/issues/50852
    - https://www.hikunpeng.com/document/detail/zh/kunpengcpfs/ecosystemEnable/Docker/kunpengdocker_03_0011.html



02 可观测落地与实践

4、业务端口与 Server 端口冲突

说明：部分 Server 端口与业务端口冲突，导致业务无法部署。可在 server 配置中可自定义各个模块所用端口：

- https://github.com/deepflowio/deepflow/blob/main/server/server.yaml

```
controller:
  ## controller http listenport
  #listen-port: 20417
  #listen-node-port: 30417
  ## grpc server port
  #grpc-port: 20035
  #ssl-grpc-port: 20135
  #agent_ssl_cert_file: /etc/ssl/server.key
  #agent_ssl_key_file: /etc/ssl/server.pem
  #ingester-port: 20033
  #grpc-node-port: 30035
  ## controller listening port
  #controller-port: 20035

  ## stats collect interval(unit: s)
  # stats-interval: 10

  ## The listening port used by Ingester to receive data
  #listen-port: 20033
  #
  ## The listening port used by Ingester to handle datasource API
  #datasource-listen-port: 20106
```

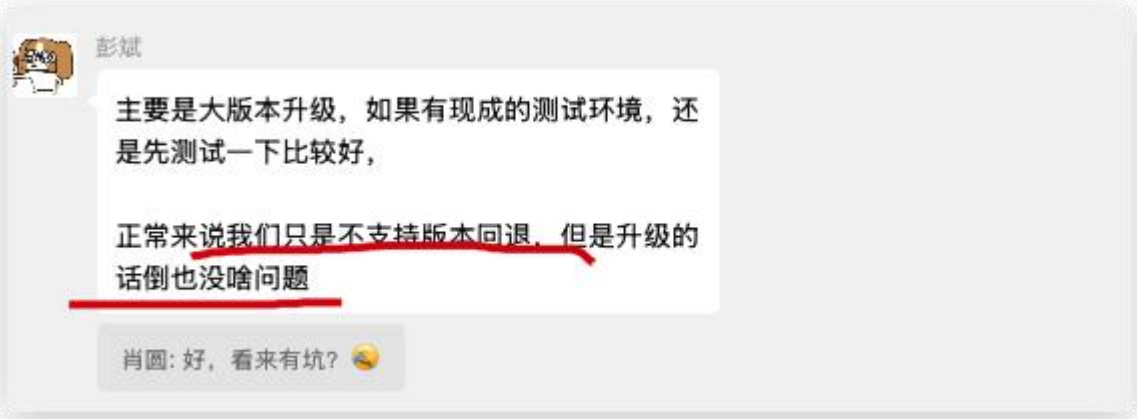
02 可观测落地与实践

5、大版本回退闭坑

说明：大版本升级的注意事项

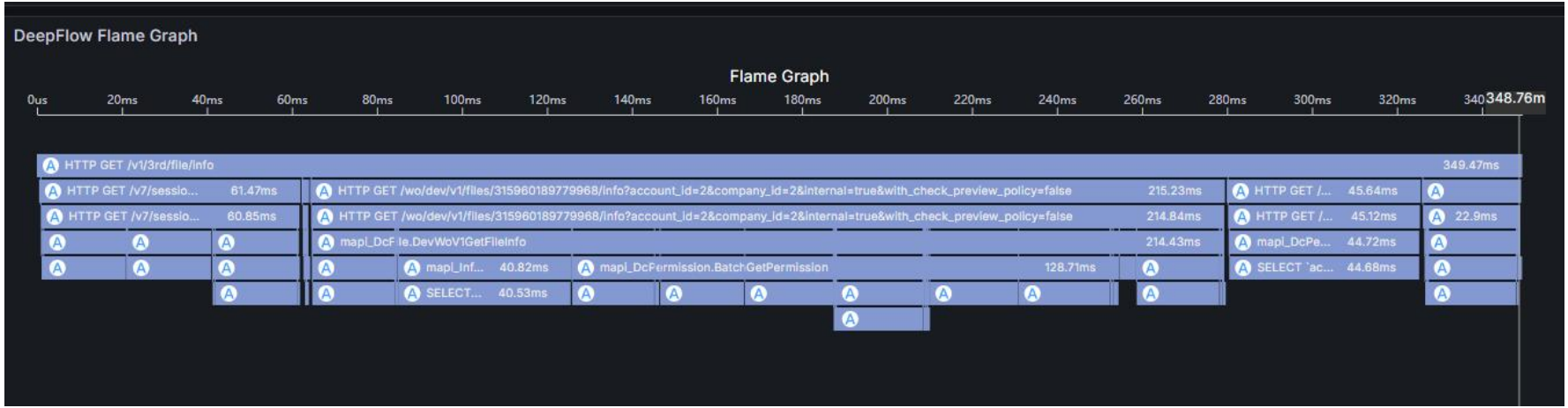
1、社区不同大版本 server 和 agent 不可兼容

2、大版本升级前一定要做测试，因为升级后如果有问题不能支持回退



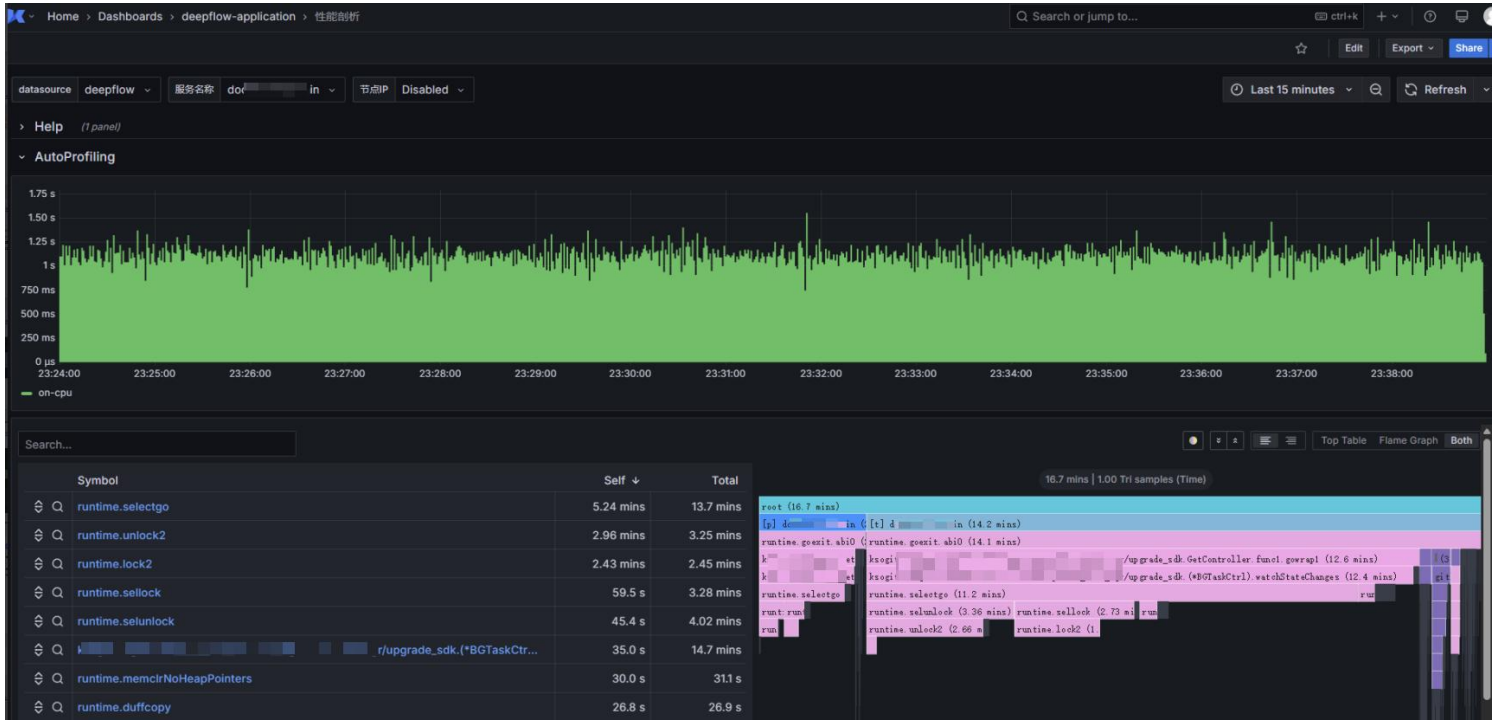
02 可观测落地与实践

阶段性成果(一)：函数级链路数据



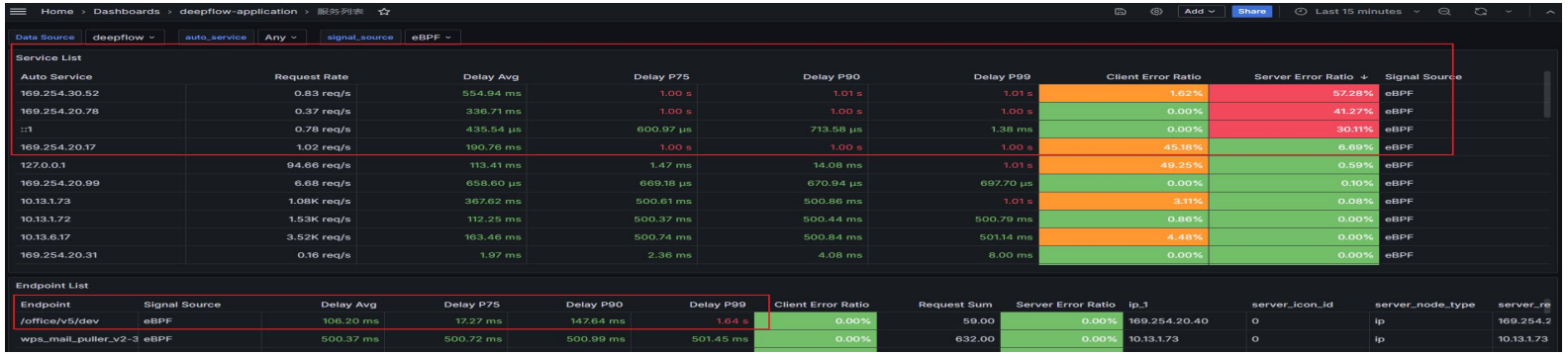
02 可观测落地与实践

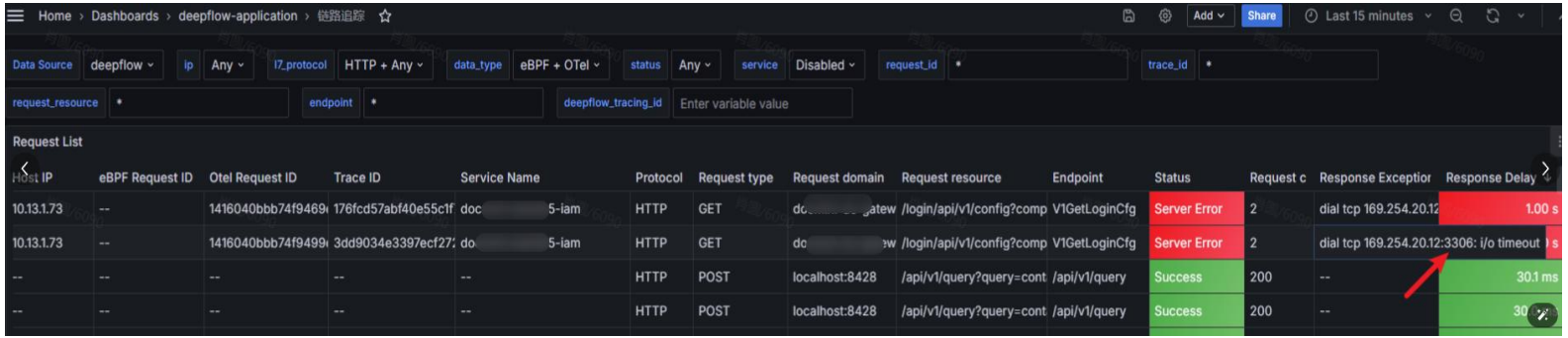
阶段性成果(二)：案例-微服务占用cpu不释放



02 可观测落地与实践

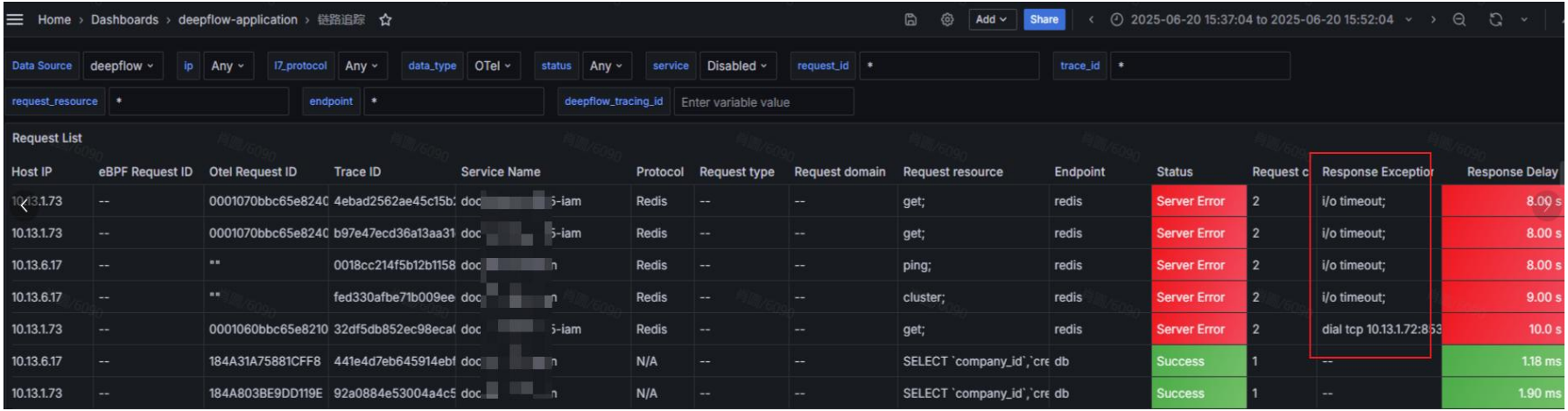
阶段性成果(三)：案例-MySQL访问延迟





02 可观测落地与实践

阶段性成果(四)：案例-Redis访问延迟



Host IP	eBPF Request ID	Otel Request ID	Trace ID	Service Name	Protocol	Request type	Request domain	Request resource	Endpoint	Status	Request c	Response Exceptior	Response Delay
10.13.1.73	--	0001070bbc65e8240	4ebad2562ae45c15b	doc[redacted]-iam	Redis	--	--	get;	redis	Server Error	2	i/o timeout;	8.00 s
10.13.1.73	--	0001070bbc65e8240	b97e47ecd36a13aa31	doc[redacted]-iam	Redis	--	--	get;	redis	Server Error	2	i/o timeout;	8.00 s
10.13.6.17	--	""	0018cc214f5b12b1158	doc[redacted]	Redis	--	--	ping;	redis	Server Error	2	i/o timeout;	8.00 s
10.13.6.17	--	""	fed330afbe71b009ee	doc[redacted]	Redis	--	--	cluster;	redis	Server Error	2	i/o timeout;	9.00 s
10.13.1.73	--	0001060bbc65e8210	32df5db852ec98eca	doc[redacted]-iam	Redis	--	--	get;	redis	Server Error	2	dial tcp 10.13.1.72:853	10.0 s
10.13.6.17	--	184A31A75881CFF8	441e4d7eb645914ebf	doc[redacted]	N/A	--	--	SELECT `company_id`,`cre	db	Success	1	--	1.18 ms
10.13.1.73	--	184A803BE9DD119E	92a0884e53004a4c5	doc[redacted]	N/A	--	--	SELECT `company_id`,`cre	db	Success	1	--	1.90 ms

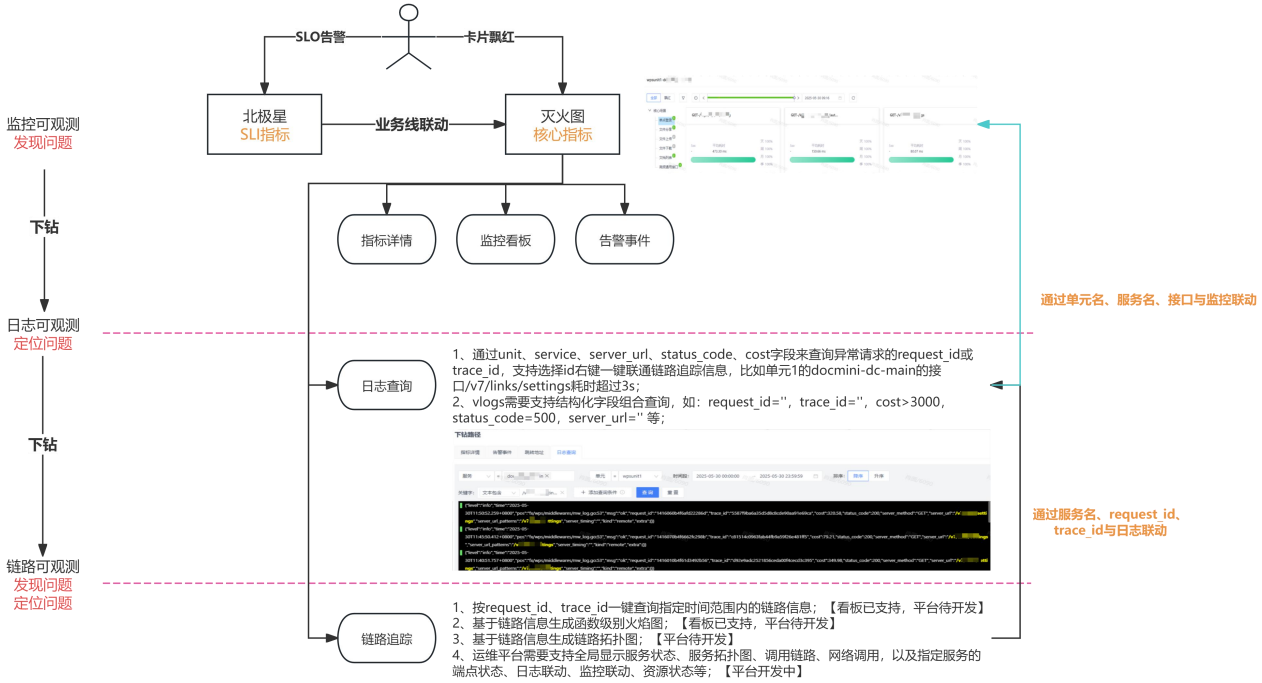
02 可观测落地与实践

阶段性成果(五)：系统适配

- 操作系统：centos、redhat、ubuntu、suse、uos、kylin
- cpu架构：amd64、arm64
- IP栈：ipv4、ipv6

02 可观测落地与实践

阶段性成果(六)：数据联动



CONTENTS

03 可观测后续规划

1、应用侧全链路接入

- 推动各业务线完成OpenTelemetry标准化接入改造
- 建立接入进度跟踪机制，确保Q3 100%覆盖核心业务系统

2、统一观测平台升级

- 开发新一代可观测分析平台，实现四维数据联动：
  - Metrics指标分析
  - Logs日志检索
  - Traces调用链追踪
  - Profiling性能剖析
- 构建多维度数据关联分析能力

3、有效性验证体系

- 设计混沌工程实验方案
- 通过故障注入验证观测系统的：
  - 指标覆盖完整性
  - 告警准确率
  - 根因定位效率

4、引入AIOps能力

- 部署智能分析Agent实现：
  - 7×24小时自动化巡检
  - 异常模式智能识别
  - 故障根因辅助分析
- 建立观测数据知识图谱

THANKS

感谢您的观看

Thank you for watching

肖圆　金山办公