夜莺监控

产品白皮书

v2019.1

www.didiyun.com

版权声明

注意

由于产品版本升级或其他原因，滴滴云将不定期对本文档内容进行更新或修正，但滴滴云不会另行通知，您可以联系滴滴云来获取最新版本文档。

本文档仅作为使用指导，文档内容可能包含技术上不准确或与产品功能及操作不相符的地方，以滴滴云最终解释为准。本文档中的所有陈述、信息和建议等均不构成滴滴云任何明示或暗示的担保、保证。

北京小桔科技有限公司

热线：400-0590-666

邮箱：nightingale@didiglobal.com

目录

产品介绍 **4**

产品优势 **4**

功能概述 **6**

功能详解 **7**

监控数据采集 7

基础指标 *7*

进程采集 *8*

端口采集 *8*

内网监控 *8*

日志采集 *8*

监控插件 *9*

监控数据存储 9

监控对象管理 9

监控看图 10

监控大盘 10

告警策略配置 11

策略继承 *12*

节点排除 *12*

标签筛选 *12*

与条件支持 *12*

告警收敛 *13*

告警升级 *13*

生效时间 *13*

留观时长 *13*

静默恢复 *13*

告警事件管理 14

告警发送 *14*

告警回调 *15*

告警认领 *15*

告警合并 *15*

告警展示 *16*

告警屏蔽配置 16

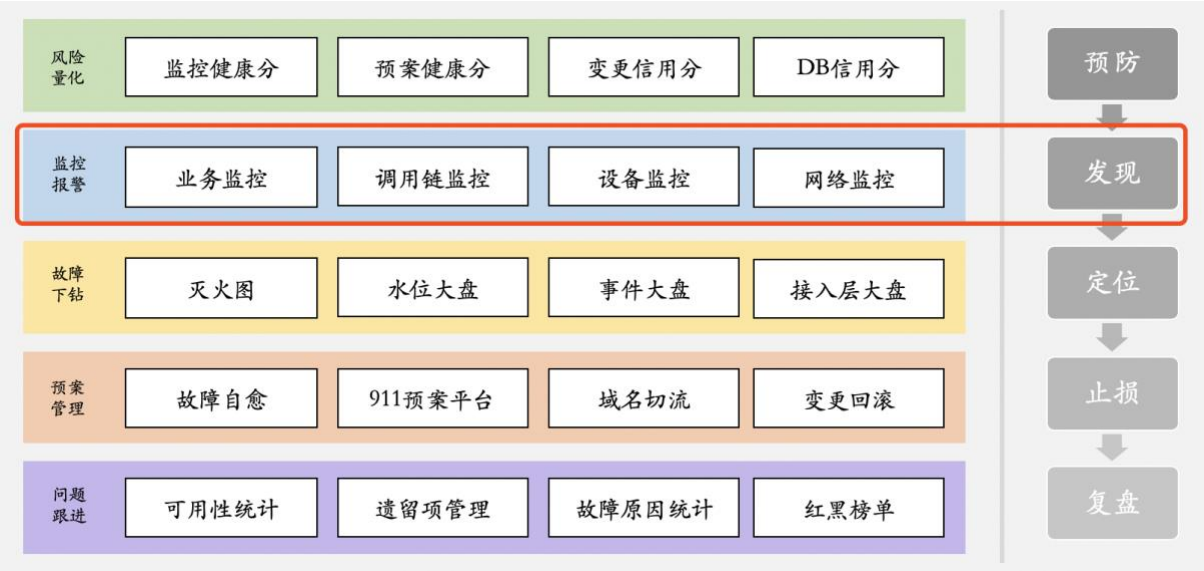
部署架构 **16**

售卖牌价 **17**

产品介绍

夜莺是一套企业级运维监控解决方案，是稳定性建设的有力保障手段。从整个故障的生命周期来看，及时发现故障，才能进行后续的止损动作，才能更好的提升服务可用性，发现故障及时报警正是监控系统的职责所在，其重要性不言而喻。

在运维稳定性体系中，滴滴有非常全面的对外商业化产品，监控报警系统在体系中的位置如下



夜莺监控能够正常工作，需要人员、权限、机器、服务树等功能模块的支持，即依赖UIC和HSP两个基础产品。这两个产品的相关信息请查看对应白皮书。

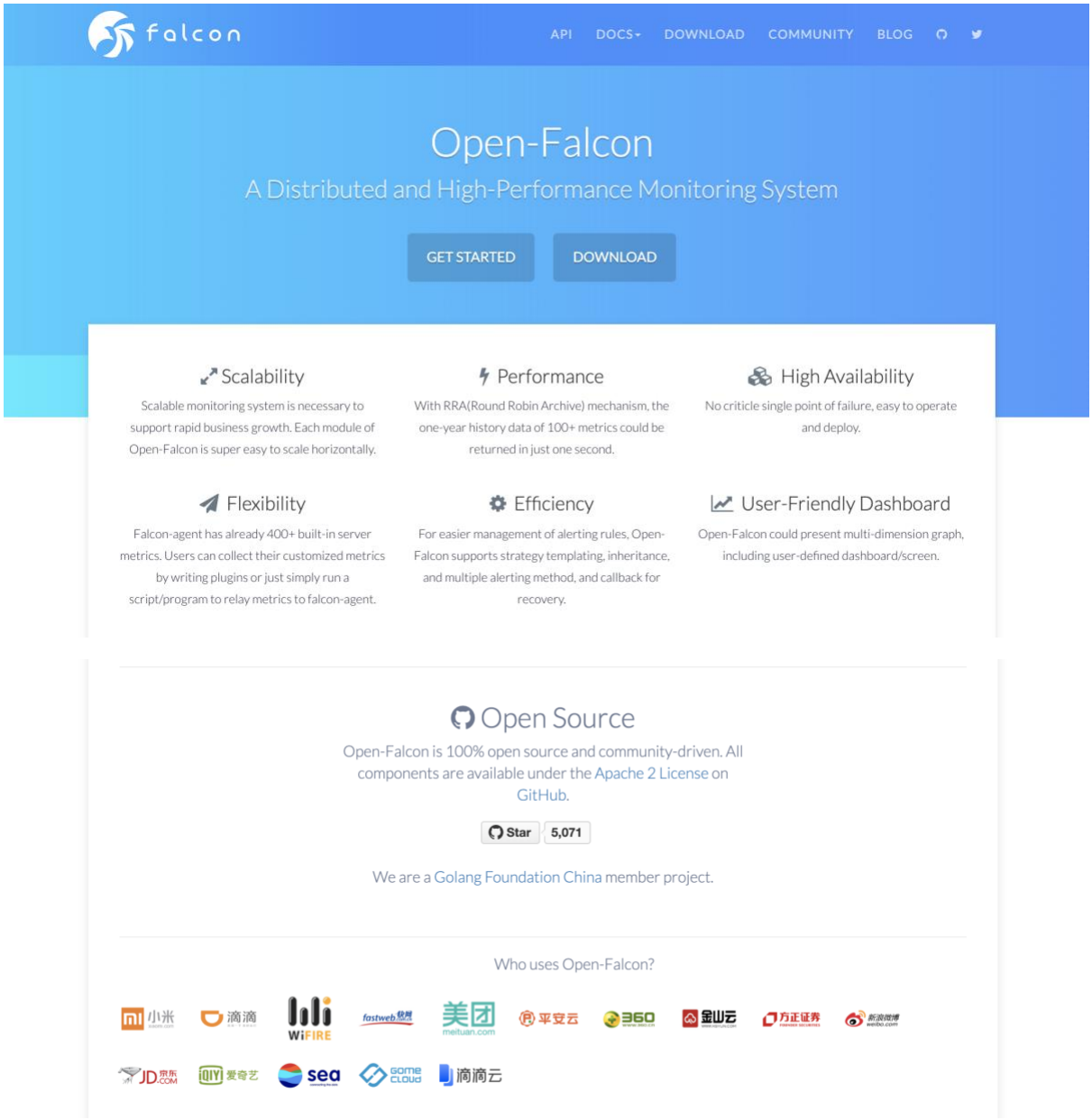
产品优势

经受住了滴滴生产考验

- 分布式、可扩展、高可用的架构设计，数据链路支持每秒**千万量级**的指标采集和处理

- 告警引擎支持**5万+**监控策略的分析和处理
- 支持管理十万量级的设备，支持**10亿**量级监控指标的存储、检索和处理
- 每周活跃用户有**5千+**的研发和运维人员，具有良好的用户体验

得到社区的广泛认可



- 我们维护着国内广泛使用的开源运维监控软件Open-Falcon，在Github获得**5000**多个star，有**80**多位代码贡献者，有美团、平安云、金山云、滴滴、360等超过**200**家商业公司在用

- 夜莺是Open-Falcon的衍生，融入了滴滴的最佳实践，重新设计了索引、告警引擎、数据存储模块，与服务树、机器管理、权限体系整合，具备更好的性能、更完备的功能、更好的易用性

功能概述

下面表格罗列了夜莺监控的各个功能，后面会围绕监控数据的整个生命周期，来详细阐述。

功能分组	功能列表
监控数据采集	硬件类：服务器、交换机、路由器、防火墙等 中间件类：MySQL、Redis、MQ、MongoDB、Oracle、PG等 大数据类：Hadoop、Spark、Storm、Flink、ES等 应用类：Nginx、JVM、Docker、应用接口、进程端口等 容器生态：K8S集群组件、容器监控 内网监控：HTTP/HTTPS/ICMP/TCP等方式探测目标 插件：系统默认没有采集的数据可以通过自定义监控插件实现能力扩展
监控数据存储	近期数据缓存在内存里，提升查询速度，采用DoD压缩算法 长期数据落盘，永久保存，有归档策略，支持存储数年的采样数据 支持双写容灾，如果成本允许可以配置为双写模式
告警数据查看	查看单个设备（比如物理机、虚拟机、交换机、容器）的监控数据，同时查看多个设备的监控数据，基于对象树的节点查看，配置监控大盘，折线图、单值、表格等多种形式看图，可以配置监控大盘用于日常巡检和排障，监控大盘支持配置阈值

告警策略配置	策略继承、节点排除、标签筛选、与条件联合告警、告警分级、告警收敛、告警升级、生效时间灵活可配、告警回调、留观时长、静默恢复等
告警事件管理	告警事件认领，邮件、短信、电话等途径发送告警事件、告警回调、未恢复告警展示、历史告警查询统计
告警屏蔽管理	告警屏蔽，可以细化到标签粒度，屏蔽原因设置、屏蔽时长设置，解除屏蔽等

监控系统主要核心功能有：数据采集、数据存储、数据展示、告警引擎、告警事件处理这五大功能。下面我们就围绕监控数据的整个生命周期，来描述夜莺监控的相关功能。

功能详解

监控数据采集

夜莺监控支持对各类对象的监控，比如物理服务器、虚拟机、容器、交换机，比如进程、端口、日志，比如数据库、缓存、消息队列等各类中间件，也支持监控插件和数据接收接口，可以说，只要能把监控数据组织成系统要求的格式，就可以接入夜莺监控。

基础指标

这里说的基础指标，是指CPU利用率、磁盘利用率、磁盘IO利用率、内存利用率、网卡流量、进程数量等等，一般监控系统都可以采集到的Linux常见性能指标，夜莺监控提供一个agent来采集这些常见指标，无需任何配置，只要部署了这个agent，这些指标就会自动采集到并且上报给监控server端。

进程采集

进程采集功能也是agent默认支持的，具体采集哪个进程，需要用户做配置，支持页面上配置，也支持在机器的指定目录放置标识文件，agent会自动读取标识文件来采集指定的进程。这个功能主要用来监控进程是否存活。

端口采集

端口监控功能也是agent默认支持的，具体采集哪个端口，需要用户做配置，支持页面上配置，也支持在机器的指定目录放置标识文件，agent会自动读取标识文件来采集指定的端口。这个功能主要用来监控端口是否存活。

内网监控

支持对目标做探测，支持的协议有HTTP、TCP、ICMP等，比如贵公司规范，所有HTTP服务都有一个/healthz接口，请求之，返回ok，据此，可以对所有HTTP服务做周期性/healthz接口探测，比进程监控和本机的端口监控都更有效，因为进程如果卡住了，从OS层面来看，进程和端口都还在，但是就是不能提供响应。远端HTTP接口探测，是可以探测到这种情形的。

日志采集

日志采集是指通过配置特定的正则表达式，从日志中提取信息，得到监控数据的方式。夜莺不会把目标机器的日志都收集到中心端，去做正则匹配，而是把要匹配的正则表达式下发到目标机器的agent，由目标机器的agent对日志文件做匹配。比如贵司有个对象存储的服务，每收到一个下载请求就会打印一条日志，注明时间、下载的文件、所属的bucket、文件大小、处理时间等等，这种日志就非常适合用正则去匹配得到监控数据，比如我们可以很轻易的统计某个bucket的qps是多少。

监控插件

由于默认提供的监控agent无法满足所有采集需求，所以，夜莺提供了两种扩展机制：一种是监控系统server端提供接口，允许用户上报监控数据，另一种就是插件机制，允许用户编写一个可执行文件去采集监控数据，一般都是脚本形式，比如shell、perl、python、ruby等脚本，用go、c等语言写插件也没问题，只要最终是一个可执行文件，放到指定的目录，用特定的命名规范，夜莺就能识别到。像是MySQL、Redis、MongoDB、MQ、ES、JVM等的监控数据采集，通常都是使用监控插件。

监控数据存储

夜莺中有一个模块叫tsdb，专门用来存储监控数据。由于监控数据的价值随着时间锐减，越是新数据，看的频率越高，越是老数据，看的频率越低，所以，tsdb会把新数据缓存在内存里，老数据落盘存储。内存里可以存多久的数据是可以配置的，硬盘上持久存储的数据，由于存放时间比较长，比如一般公司可能会存一年的数据，tsdb会对数据做归档处理，底层使用rrdtool的格式来存储，所以天然具备采样归档能力。

监控对象管理

首先解释一下何为监控对象。顾名思义，就是指我们要监控的那个对象，这个对象会有很多指标来体现它各个维度的信息。比如某个物理服务器，就是个监控对象，它有CPU相关指标，有内存相关指标，有磁盘相关指标；比如某个交换机，某个虚拟机，某个容器，都是监控对象；各种中间件，比如某个MySQL实例，某个Redis实例，都是监控对象；研发人员写的业务进程也是监控对象。有些指标所属的对象是整个集群的，比如某个CEPH集群的整体容量水位，此时，这个CEPH集群就是监控对象。

监控数据上报的时候，是通过endpoint字段指明所属监控对象的。从监控server端来看，会收到很多endpoint（即监控对象）的各种维度的监控数据上来。此时一个最典型的需求就产生了：对endpoint分组。比如看监控数据，不可能一次性看所有endpoint的，常见场景是，某个服务部署到了5台机器上，为了看这个服务对机器资源的使用情况，我们会同时看这5台机器的监控数据；比如配置报警策略，也不可能同时对所有endpoint配置完全相同的报警策略，也是要对endpoint分组，比如分组a配置8080端口的报警策略，分组b配置了9090端口的报警策略，诸如此类。

为了与体系更好的整合，监控对象的分组借助HSP的服务树实现，HSP的产品介绍可以参看其产品白皮书。

监控看图

告警看图的功能说起来较为简单，可以查看单个设备的指标，也可以同时查看多个设备的指标，多个设备的指标展示到一张图里，会有多条线，可以选择聚合函数来聚合这多条线的数据，比如求最大值、求和、求平均值等等。

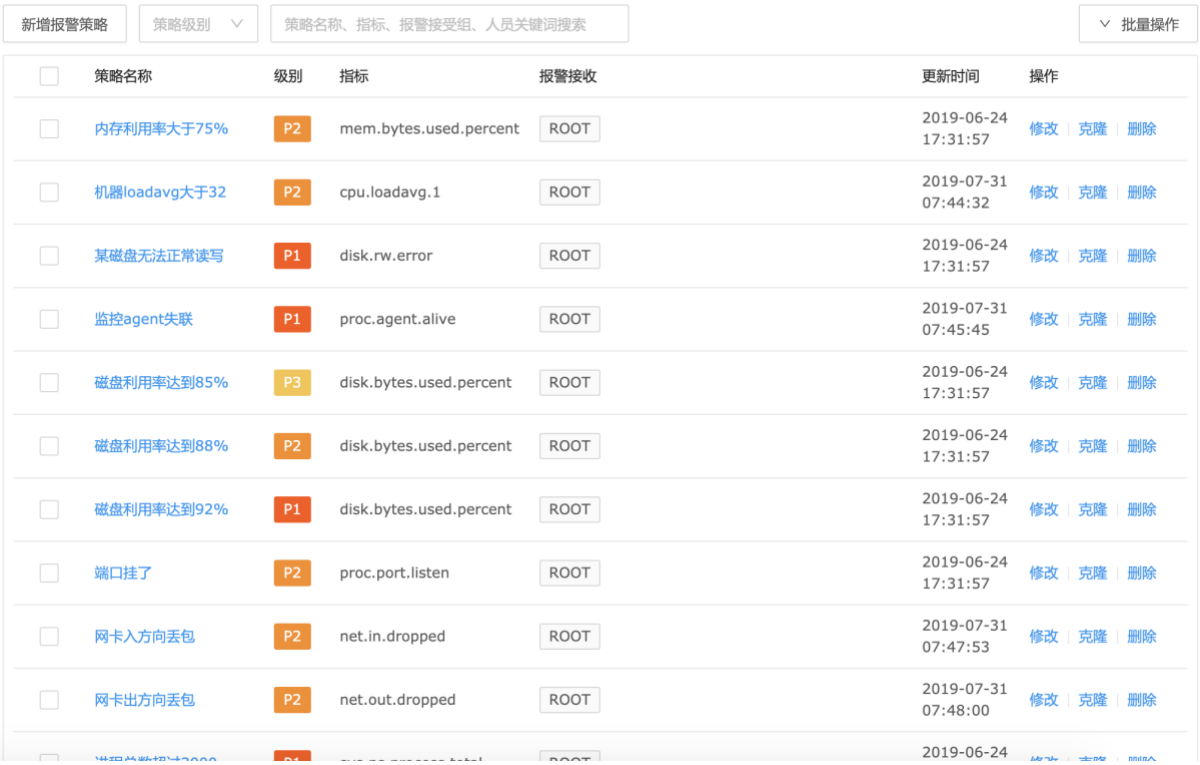
监控大盘

监控大盘是把一些经常看的图表提前配置到一个页面里，每次打开这个页面就可以立马看到这些图。监控大盘的应用场景主要是日常巡检、问题排查、趋势分析。

由于公司会有很多部门、很多业务，所有监控大盘一股脑罗列在一个表格里，搜索查看都会比较麻烦，所以我们是把监控大盘和服务树关联在了一起，因为服务树天然就描述了组织架构和服务模块关系，是我这个服务的监控大盘，我就让它挂到我的服务节点上去，便于管理。

监控大盘里的每张图，都可以自定义标题，可以随意拖拽调整顺序，可以配置阈值红线作为标识，这样可以很轻易的知道哪些指标达到了危险区域需要关注。这里给一个监控大盘的样例供参考：

告警策略配置



夜莺监控策略的配置非常灵活，从真正工作中的实践总结而来，当然，对于新手可能会觉得有些复杂，下面挨个介绍各个功能特色。

策略继承

告警策略是配置到某个树节点上的，这个树节点的所有子节点、孙节点就都会继承这个策略，子孙节点上面挂的那些设备，就自动应用了这些告警策略，大幅减少告警策略的配置条数，提升可维护性。

节点排除

由于告警策略是绑定在某个树节点的，这个节点的子孙节点都会继承父节点上面的告警策略。那如果某个子孙节点比较特殊，不希望继承父节点的告警策略，怎么办？夜莺支持节点排除配置，在配置告警策略的时候，可以把部分子节点排除掉，不应用父节点的告警策略。当然，从实践角度，还是要通过良好的树结构组织，尽量避免这种情况，会更易于维护。

标签筛选

比如监控硬盘利用率，我们只想监控根分区和/data分区，不想监控/boot分区，此时可以通过标签筛选里边的排除功能，排除掉/boot分区，当然，也可以使用标签包含功能，指定只监控根分区和/data分区。

与条件支持

夜莺支持配置两个条件都满足才报警，比如disk.io.util大于99%，同时还要求disk.io.avgqu_sz大于32才报警。

告警收敛

支持在指定时间内最多报几次。比如一般配置一小时内如果持续触发阈值，最多报2次，不会一直报。下一小时如果还没恢复，还会报2次，以此类推。

告警升级

如果某个报警报出之后一直没有人处理，而且告警也一直没有恢复，可以触发告警升级，发告警给老板，这样一来，告警的跟进处理速度会明显加快，有助于大幅缩减服务不可用时长。

生效时间

告警策略可以配置灵活的生效时间，比如只有工作日的9点到20点生效。可以利用这个机制，对同一个指标，不同时间段不同阈值。

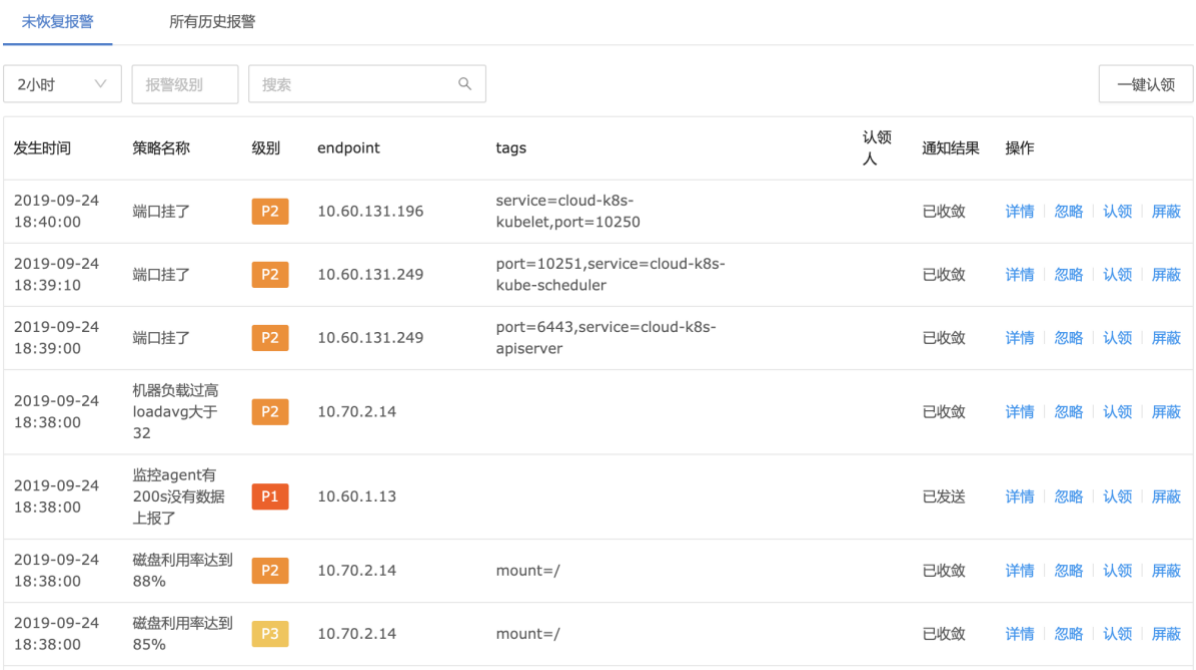
留观时长

比如cpu.idle我们配置的告警策略是小于10%就告警，某机器负载很高，持续性的维持在5%左右，所以肯定会报警，但是偶尔有一次回到了20%，然后下一周期又继续维持在5%，对于这种情况，20%的那次已经大于我们的告警阈值，所以正常来说，告警恢复了，但实际这只是个短暂毛刺现象，应对这种情况，我们可以配置留观时长，就像病人留院观察一段时间，确实恢复了才认为是真恢复。

静默恢复

默认情况，触发阈值了会发告警消息，恢复了会发恢复消息，但是有时，我们不希望接收恢复通知，只接收告警通知，此时可以勾选静默恢复，在恢复的时候也就不会发送恢复通知了。

告警事件管理



告警策略配置完成之后，如果有异常监控数据产生，就会触发告警策略，生成告警事件，告警事件后续有哪些管理动作呢？本章详细阐述

告警发送

告警事件产生之后，最常见的做法是发送给相关人员，相关人员接收到报警就可以及时介入处理。告警发送的常见方式是IM消息、邮件、短信、电话，由于各个公司使用的IM不同，采集的短信电话厂商不同，导致API各异，没法统一处理。此时需要有一些适配工作，夜莺支持两种接入适配方式，脚本和HTTP接口，比如您选择HTTP接口方式，那么就需按照夜莺要求的接口规范编写HTTP接口，系统要发送短信的时候，就调用您提供的接口即可。邮件是有统一的SMTP协议来发送的，所以夜莺默认就提供了支持，只要您做好SMTP相关配置，系统就会自动发出告警邮件。

告警回调

告警回调是一种自动化手段，可以在告警事件触发的时候，调用您的某个HTTP接口，把告警事件推送给这个接口，您就可以在这个接口实现里编写相关处理逻辑。

比如我们可以用这种方式实现告警自愈。找一个中控机，上面部署ansible，打通与目标机器的信任关系，编写一个web服务部署到中控机上，提供HTTP回调接口配置到监控里，告警事件触发之后回调您的HTTP接口，您就可以知道是哪个机器哪个监控指标告警了，然后利用ansible去目标机器run一个shell脚本，以达到自愈的目的。举个例子：某机器的crond进程挂了，触发了告警，回调到这个HTTP接口，这个接口的处理逻辑一看是crond挂了，那就去目标机器执行systemctl start crond，搞定。

夜莺的这个能力可以与任务中心打通，通过任务中心来管理所有任务，报警之后自动去目标机器执行指定的脚本，非常方便。

告警认领

如果您的告警策略配置了告警升级机制，告警事件触发之后，假设长时间未恢复，就会自动升级到您的老板，如果我们已经在处理这个告警，那就无需升级了，怎么告诉系统您已经在介入处理了呢？就是告警认领机制，在未恢复的告警页面，可以看到每条告警后面都有一个认领按钮，点击，即代表您认领了这条告警，这个告警就不会往上升级了。

告警合并

如果出现某个底层故障，比如网络问题，可能上面依赖的众多服务都会报警，此时容易产生告警风暴，此时告警合并就派上用场了，夜莺会对低优先级（可以配置何为高优先级告警，何为低优先级告警）告警做合并处理，减少告警发送的条数，减少对告警接收人的打扰。

告警展示

告警历史页面，可以看到未恢复的告警和历史所有告警，这个未恢复的告警要额外关注，作为早晚巡检的必看页面，以防漏掉告警，所有历史报警是个留底，用于后面的统计分析，产生报表，指引我们告警优化方向。

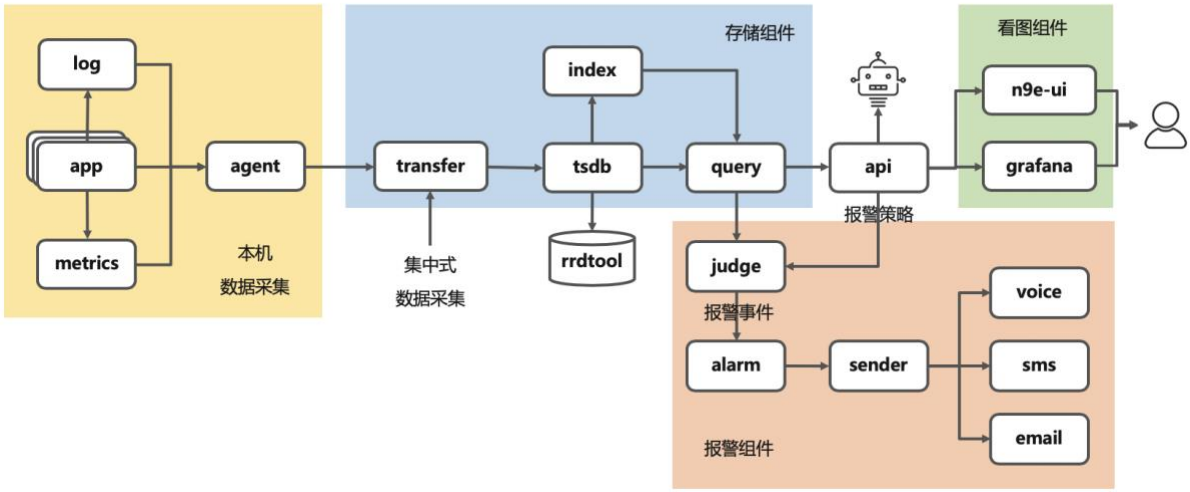
告警屏蔽配置

对于一些有预期的变更，必然会触发告警的情形，我们可以提前配置屏蔽策略，省的触发告警之后打扰到大家。比如典型的机器重启，这个一般是P1电话告警，为了让大家免受摧残，可以在机器重启之前先屏蔽这个机器的告警。

告警屏蔽支持配置屏蔽时间，支持填写屏蔽理由，支持屏蔽某个设备的某个指标，也可以细化到屏蔽某个设备的某个指标的某个标签。实践的时候，屏蔽时间一般不要设置过久，够用就好，否则如果忘记解除屏蔽，就会漏掉一些重要告警。

部署架构

本节对夜莺监控系统的部署架构做简要介绍。下面架构图是数据流向图：



- 黄色底色的部分是agent，部署到所有目标机器，用于采集监控数据。除了agent默认采集的基础指标，还可以通过分析日志文件的方式提取监控指标，最后将数据推给transfer
- 蓝色底色部分是数据存储组件，用来接收数据，存储数据，建立索引，提供查询接口
- 橙色底色部分是报警组件，从api组件拉取监控策略，从query组件拉取监控数据，在内存里做比对，做报警判断，生成的告警事件写入redis，由alarm来消费，告警发送是由sender模块负责
- 绿色底色部分是看图的UI，系统内置了看图组件，当然，也可以自己搭建grafana来做图表展示

除去黄色底色部分，剩下的全部是服务端的组件，为了容灾考虑，我们推荐至少使用3台机器，由于监控数据的落盘是IO密集行为，尽量使用SSD机器。所以，最小的部署要求：

CPU	内存	硬盘
8	16	300G SSD

这样配置的机器，单台机器每秒大约可以承载15K监控指标，3台机器就是45K，如果监控数据采集周期是20秒，每个周期可以承载90万指标，只是用于设备监控的话，预计可以承载**3000台**设备的监控需求。服务端各个组件支持扩容，可以通过部署更多服务端组件来监控更多设备。

售卖牌价

我们提供两种售卖方案：

方案一：附源码，没有license限制，300万RMB

方案二：无源码，根据管理的设备数量做梯度售卖

设备数量	价格
[0-100)	70万RMB
[100-500)	90万RMB
[500-1000)	130万RMB
[1000-2000)	160万RMB
[2000-4000)	190万RMB
[4000-6500)	220万RMB
[6500-10000)	250万RMB

对于方案二，如果需要售后维保服务，维保费用是合同金额的25%，第一年免费维保，从第二年开始收费，具体维保内容请参看维保服务相关说明文档。