

Please visit website: <http://cxyroad.com>

国产监控之光 nightgale，还得是开源社区呐！

=====

笔者前段时间机缘巧合下，成为了 [nightgale](<http://cxyroad.com/https://github.com/ccfos/nightingale>) 的贡献者，之后会分享一些关于参加开源的一些经历，*个人同名公粽号：Ciusyan*，欢迎志合者一起交流啊~

* 项目 Github 链接: [github.com/ccfos/night...](<http://cxyroad.com/https://github.com/ccfos/nightingale>)

* 官方文档: [flashcat.cloud/docs/conten...](<http://cxyroad.com/https://flashcat.cloud/docs/content/flashcat-monitor/nightingale-v7/introduction/>)

![star-history-202471.png](<https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a47d718fe9cd4b99aa2f314be92cf2de~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1832&h=1308&s=124377&e=png&b=ffff>)

既然是刚开始，先简单介绍一下，以我的视角，带大家看看这一：**国产监控之光**，是怎样一回事儿吧！

0x00 我眼中的监控？

=====

为什么说他是国产监控之光呢？先不着急回答这个问题，先来看看，什么是可观测性领域的监控吧！

相信下图大家都不陌生，从小到大。我们无不被各种监视，从人工时刻看管，到借用电子设备监测。

对于传统的一些监控设备，核心是拿来监控某一实时情况的，比如用于安全监控、交通监控，以此让我们的生活变得安心放心一些。

那么对于我们技术界中，服务的监控是什么呢？其实与也有异曲同工之妙，核心就是对一些数据进行监控，以此来让我们负责的服务安心一些。

为什么这么说呢？当自己实习了近一年后，能深刻的体会到，如果出现线上事故：

- * 对于公司：不能及时响应，稍不注意就会有很大的损失。
- * 对于个人：不能及时处理，稍不注意就会有被优化的风险。

所以这就体现了监控的重要性，想要第一时间发现问题，**那就必须得将其核心指标监控起来，并且配置上合理的告警**。如果一个在线系统，没有建设监控或者监控指标不完善时，一旦发现问题，可能就已经是有很大的损失了。举个：

假设有一个发放优惠券的功能，当系统中有一些未知的 bug，如果导致了优惠券超发。如果对于这些发放的券没有做一些兜底或者监控，可能发现异常的时候，都已经发出了很多异常券。那么这对于这一业务，就是一重大事故和损失。

但如果对对该券配置一些监控告警，在发现发出的优惠券超出了预期后，能够第一时间响应，就可以更快速的止损。

简单说了一下我对于监控的理解，那我为什么说夜莺是国产监控之光呢？可能大家一提到可观测性，就会想到 Prometheus，它的确是一个非常优秀的开源项目，并且在现在的可观测性领域，很多标准都是与 Prometheus 对齐的。

但其实纵观大多数公司，如果完全自己建一套这样的可观测性系统，显然会付出巨大的成本，即使是使用开源的产品，也需要投入巨大的人力成本。那么有没有这样一个监控产品，**能满足企业的可观测性需求，又能以最低成本接入呢**？这就是夜莺开源版存在的意义。

当然，上述纯属个人简单观点，可否认，不反驳。那么在我眼中的夜莺，是什么样的呢？

0x01 我眼中的夜莺？

=====

先来看看它官网和各大博客中随处可见的介绍：

夜莺是一个 All-in-One 的云原生监控工具，**集合了 Prometheus 和 Grafana 的优点，** 可以方便在 WebUI **界面管理和配置告警策略**，也可以对分布再多个 Region 的指标、日志、链路追踪数据进行统一的**可视化分析**。

夜莺是一个服务端组件，类似 Grafana，可以接入不同的数据源，夜莺就可以对数据源的数据进行分析、告警、可视化，以及后续的事件处理、告警自愈（和 Grafana 一样提供可视化，对告警规则管理不同于 Prometheus 通过配置文件来实现，夜莺通过 WebUI 来统一协同）夜莺商业版产品提供了更多企业级功能，用于统一监控和故障定位场景。

上面的介绍比较官方，那么我眼中的夜莺是什么呢？在扒官方文档的时候，看到条很核心的指导思想： **“夜莺不生产数据，只是日志‘监工’”。**

所以，我眼中的夜莺，可以理解为**它是将「日志、指标」等信息，通过可视化的方式，聚合展示至 WebUI 的一个工具，并侧重于对其数据的利用。**

那么就有几个很核心的点：

1. 日志记录，指标上报
2. 日志采集，入数据源
3. **分析日志，监控指标**
4. **配置告警，及时响应**

而夜莺呢？核心是在后两个点。也就是说，通过某种手段（采集），将记录的日志和上报的指标采集进入某些数据源后（时序库、索引库），可以使用夜莺来对「时序指标、日志」进行分析展示，可视化至 WebUI。

最后总结一下，我眼中的夜莺是什么呢？**对日志精细化利用（一切数据皆为日志），将用户关心的内容、数据进行可视化展示及其监控。并且为其提供丰富的告警能力，方便及时发现问题，快速定位问题，快速解决问题。**

0x02 有哪些模块?

=====

对于上述有了一个基础印象之后，我这里就不介绍如何快速部署安装入门了，官网又很详细的文档，可自行查阅~

那么夜莺有哪些模块呢？我们可以看看它的 UI 界面。

从这里，可以了解到夜莺提供的基础功能，其大致介绍可参考[夜莺功能概览](http://cxyroad.com/ "https://flashcat.cloud/docs/content/flashcat-monitor/nightingale-v7/overview/"). 那么什么是核心功能，夜莺的核心竞争力呢？在谈此之前，我们先来了解一个 Basic。简单扒了一下文档，我们可以将夜莺抽象为下图：

如图所示，我们可以将夜莺理解为一个中转站，它只负责打通前后「数据源、存储库」两个模块的链路。前后两个过程，是完全独立解耦的。也就是说：

1. 可以用夜莺打通这样的链路：将采集器采集的数据，存储至各类存储库（时序库、索引库），但是要不要用 n9e 进行分析展示呢？随你选择。
2. 可以用夜莺对存储库中现有的数据，在 n9e 进行分析并可视化。但需不需要关心这些数据是否是通过 n9e 配置的采集器采集来的呢？根本不关心。

当然，再重复一遍其中的核心思想：**夜莺本身不具备采集、存储能力。它只负责打通整条链路，更方便进行配置、展示、分析、管理、告警。**

了解了这样一个基本前提，那么可以将夜莺简单的理解为，有三个核心模块：

1. 打通数据源采集相关的模块

- * 插件化接入各种数据源的采集器，比如 Datadog-Agent、Telegraf、Grafana-Agent、Categraf...

- * 利用采集器去采集对应的指标、日志数据。

- * 将其数据上报至夜莺，进行加工。

2. 夜莺自身 Server 模块

- * 对 Web UI、夜莺工作台进行管理及其相关操作。

- * 加工处理数据，上报心跳...

- * 分析数据、可视化日志数据、展示大盘..

- * 接入各类三方插件

3. 打通数据存储相关的模块

- * 插件化接入各种时序库和索引库，比如 Prometheus、Victoriametrics、M3DB、Thanos、Elasticsearch...

- * 夜莺拉取数据做对应的操作

0x03 核心功能介绍

=====

根据官网，准备好一些前置组件和环境后，如何使用夜莺打通链路呢？

大致的步骤是这样的：采集器配置 -> 配置数据源 -> 时序指标、日志分析 -> 配置仪表盘 -> 告警管理。

采集器本身的配置就不在这里展开了，如果使用 categraf，官网提供了很多 input 插件，开箱即可采集很多常见组件的指标。**核心点是将采集的数据，发送至 n9e Server，由它加工后，再将其存储至配置的存储库。**

)

上面也提到了，数据推给夜莺了，夜莺加工后要发送至哪里呢？所以需要配置数据源，比如推送至时序库、索引存储引擎。

将采集的数据推送给对应的数据源后，就可以做对应的查询了：

每次都来查询好麻烦的，配置一些监控看板。

可以快速导入内置的一些大盘，有很多常见的组件都包含进去了。

也可以自己配置想要的大盘，但是不知道怎么配成上面那种很好看的，*（在界面好像配置不了，只能通过对应的大盘 Json 去描述修改）。*

当完成了上述的一些步骤，其实平时就能够根据这些大盘，查询、定位、分析很多问题了。

比如我们看到某些图表中，到达了某些阈值、有毛刺、突增突降... 说明可能是

系统哪里异常，或者哪里波动了，我们就可以通过一些指标的一些信息，去快速定位检查。

但是我们不可能实时盯着这些大盘是吧，不现实也不可能。那么应该如何操作呢？那就是让系统帮我们监视，并且有一次给我们发送告警。所以再往下面，可以配置告警管理。

0x04 写在后面
=====

好了，基本介绍了一下夜莺这一项目，后续还会更新关于夜莺的一些文章，记录一下这一开源过程。如果想学习 Golang 和入门可观测性领域的的朋友，这一项目真是倾情推荐！！！别问，问就是我就是自己试过。

欢迎大家来公粽号交流：**Ciusyan**，坚持做一名特立独行的输出者！

最后，用在官网看到秦总说的一句话来结束本篇： **“Grafana 更擅长监控面板的管理，N9e 更擅长告警规则的管理。”** 所以，告警管理的模块，一定是n9e 的核心功能，更多的开源需求和 PR 会在后续的文章中同大家一起分享。

原文链接: <https://juejin.cn/post/7385752495535423503>