|  |
| --- |
| OAMLab |
| 运维实验.夜莺 V5 服务器监控告警 |
| 部署、告警 |

|  |
| --- |
| 况波，1214966109@qq.com  2022-12-17  基于GPL v3 |

目录

[一、夜莺介绍 2](#_Toc122199719)

[二、产品对比 2](#_Toc122199720)

[三、工作原理 3](#_Toc122199721)

[四、采集器 4](#_Toc122199722)

[Categraf 4](#_Toc122199723)

[Telegraf 4](#_Toc122199724)

[Exporters 4](#_Toc122199725)

[五、TSDB 5](#_Toc122199726)

[Prometheus 5](#_Toc122199727)

[Thanos 5](#_Toc122199728)

[VictoryMetrics（VM） 5](#_Toc122199729)

[其他 5](#_Toc122199730)

[六、系统架构 6](#_Toc122199731)

[七、安装部署服务端 7](#_Toc122199732)

[1. 服务器准备 7](#_Toc122199733)

[2. 服务器基础初始化 7](#_Toc122199734)

[3. 目录初始化 7](#_Toc122199735)

[4. 安装 MySQL 7](#_Toc122199736)

[5. 安装 Redis 7](#_Toc122199737)

[6. 安装 Prometheus 7](#_Toc122199738)

[7. 安装 N9E 8](#_Toc122199739)

[8. 导入基础数据 8](#_Toc122199740)

[9. 配置启动文件并启动 8](#_Toc122199741)

[10. 安装 Nginx 代理访问 9](#_Toc122199742)

[八、安装部署采集器 10](#_Toc122199743)

[Categraf 10](#_Toc122199744)

[Telegraf 11](#_Toc122199745)

[九、平台介绍 11](#_Toc122199746)

[十、告警自定义 11](#_Toc122199747)

# 一、夜莺介绍

Nightingale | 夜莺监控，一款先进的开源 云原生监控分析系统，采用 All-In-One 的设计，集数据采集、可视化、监控告警、数据分析于一体，与云原生生态紧密集成，提供开箱即用的企业级监控分析和告警能力。

于 2022 年 5 月 11 日，捐赠予中国计算机学会开源发展委员会（CCF ODC），为 CCF ODC 成立后接受捐赠的首个开源项目。

官方地址：<https://n9e.github.io/>

快猫地址：<http://n9e.flashcat.cloud/>

从 v5 版本开始与 Prometheus、VictoriaMetrics、Grafana、Telegraf、Datadog 等紧密协同集成，提供开箱即用的企业级监控分析和告警能力。

夜莺监控，由滴滴开发和开源，其核心开发团队，也是 Open-Falcon 项目原核心研发人员。

# 二、产品对比

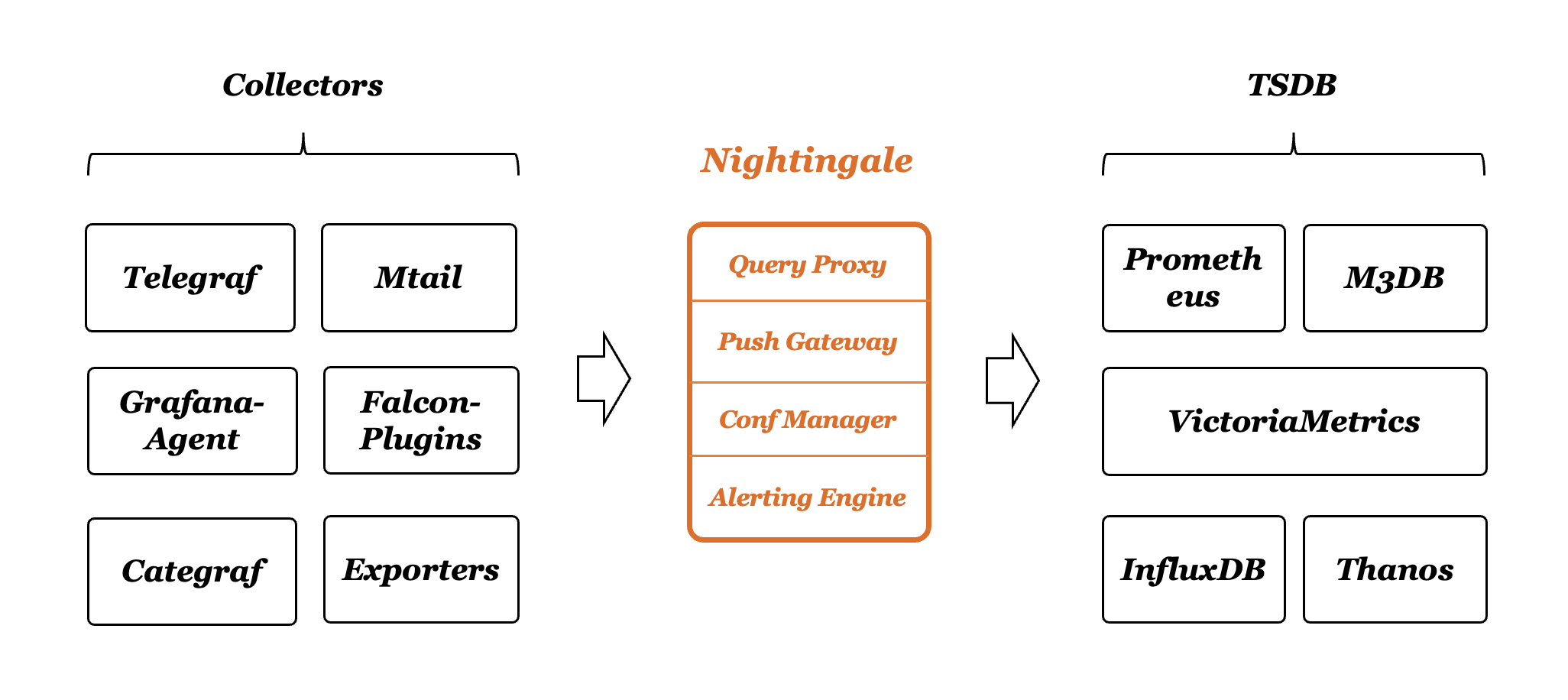
解决了 Prometheus 使用过程中的以下问题：

1. Prometheus、Alertmanager、Grafana 等多个系统较为割裂，缺乏统一视图，无法开箱即用。
2. 通过修改配置文件来管理 Prometheus、Alertmanager 的方式，学习曲线大，协同有难度。
3. 数据量过大而无法扩展您的 Prometheus 集群。
4. 生产环境运行多套 Prometheus 集群，面临管理和使用成本高的问题。

解决了 Zabbix 使用过程中的以下问题：

1. 监控的数据量太大，希望有更好的扩展解决方案。
2. 学习曲线高，多人多团队模式下，希望有更好的协同使用效率。
3. 微服务和云原生架构下，监控数据的生命周期多变、监控数据维度基数高，Zabbix 数据模型不易适配。

# 三、工作原理



# 四、采集器

### Categraf

开源地址：<https://github.com/flashcatcloud/categraf>

教程地址：<https://n9e.github.io/docs/agent/categraf/>

说明：一款 all-in-one 的采集器，由快猫团队开源，是夜莺主推的采集器。

能力：不但可以采集 OS、MySQL、Redis、Oracle 等常见的监控对象，也准备提供日志采集能力和 trace 接收能力。

特点：每个采集器都有一个目录，在 conf 下面以 input. 打头，如果不想启用某个插件，就把插件配置目录改为不是 input. 打头即可。

缺点：目录分的比较细。配置就显得相对于多一点。

### Telegraf

开源地址：<https://github.com/influxdata/telegraf>

教程地址：<https://n9e.github.io/docs/agent/telegraf/>

说明：是 InfluxData 公司开源的一款采集器，内置非常多的采集插件。

问题：不过 Telegraf 是面向 InfluxDB 生态的，采集的监控数据推给 InfluxDB 非常合适，推给 Prometheus、Victoriametrics、Thanos 这些时序库，可能会带来问题。主要问题有几个：

1. 有些数据是 string 类型的，Prometheus、VM、M3、Thanos 等都不支持 string 类型的数据。
2. 有些采集器设计的标签是非稳态的设计，比如经常会看到 result=success 和 result=failed 的标签，需要手工配置采集器 drop 掉，但是对于新手确实有些难度。
3. Telegraf 采集的数据存到 Prometheus 中，再接入 Grafana，这种使用方式 Grafana 的大盘模板比较少。

### Exporters

常用的 Prometheus 生态，如 Node Exporter 等。

# 五、TSDB

## Prometheus

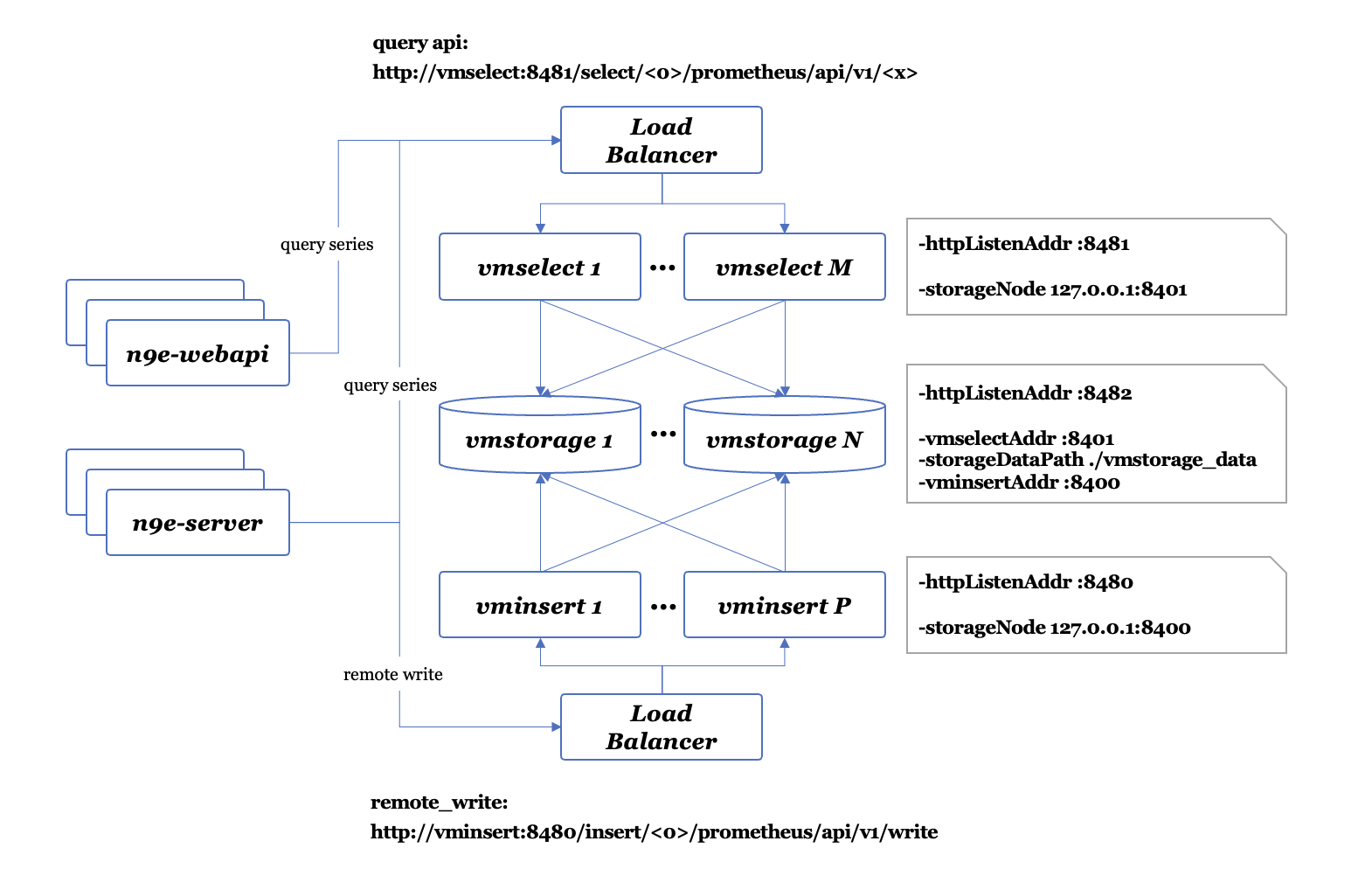
简单易用，Go 编写，本身自带了 WebUI 查询界面。但是集群部署不是很友好。

## Thanos

相当于 Prometheus 的加强版，优化了集群部署。

## VictoryMetrics（VM）

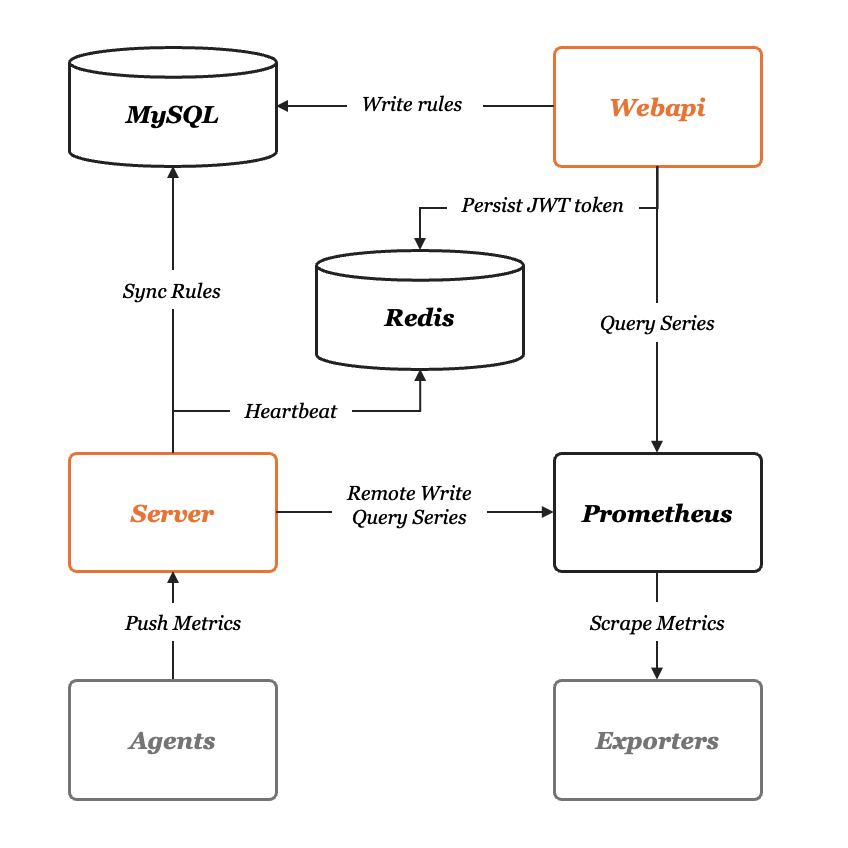
Prometheus 的完美加强版。架构简单，可靠性高，在性能，成本，可扩展性方面表现出色，社区活跃，完全兼容 Prometheus 生态。提供[单机版](https://docs.victoriametrics.com/Single-server-VictoriaMetrics.html)和[集群版](https://docs.victoriametrics.com/Cluster-VictoriaMetrics.html)。如果您的每秒写入数据点数小于100万（相当于每个机器差不多采集200个指标，采集频率是10秒的话，每台机器每秒采集20个指标左右，100万/20=5万台机器），单机版可以通过增加服务器的CPU核心数，增加内存，增加IOPS来获得线性的性能提升。且单机版易于配置和运维。[详细说明](http://n9e.flashcat.cloud/docs/install/victoria/)，简单的架构图：



## 其他

例如 M3DB，InfluxDB。

# 六、系统架构



各种环境的选型建议：

1. 快速体验测试，使用 Docker compose 方式
2. 公司大规模使用了 Kubernetes，可以选中 Helm 方式
3. 最稳定的部署方式，还是二进制
4. 小规模使用，比如 1000 台机器以下，用 Prometheus 做存储即可，超过 1000 台机器，选择 VictoriaMetrics 可能更合适

其他文档：<http://n9e.flashcat.cloud/docs/usage/share/>

# 七、安装部署服务端

## 服务器准备

准备2台测试服务器：CentOS 7.9 4G/4C/20

## 服务器基础初始化

关闭防火墙，内核调优等。

## 目录初始化

cd /opt/  
mkdir service package backup

## 安装 MySQL

yum -y install mariadb\*  
systemctl enable mariadb  
systemctl restart mariadb  
mysql -e "SET PASSWORD FOR 'root'@'localhost' = PASSWORD('1234');"

## 安装 Redis

yum install -y redis  
systemctl enable redis  
systemctl restart redis

## 安装 Prometheus

# 下载安装  
cd /opt/package/  
wget https://s3-gz01.didistatic.com/n9e-pub/prome/prometheus-2.28.0.linux-amd64.tar.gz -O prometheus-2.28.0.linux-amd64.tar.gz  
tar xf prometheus-2.28.0.linux-amd64.tar.gz  
mv prometheus-2.28.0.linux-amd64 /opt/service/prometheus  
​  
# 配置启动文件  
cat <<EOF >/etc/systemd/system/prometheus.service  
[Unit]  
Description="prometheus"  
Documentation=https://prometheus.io/  
After=network.target  
​  
[Service]  
Type=simple  
ExecStart=/opt/service/prometheus/prometheus  --config.file=/opt/service/prometheus/prometheus.yml --storage.tsdb.path=/opt/service/prometheus/data --web.enable-lifecycle --enable-feature=remote-write-receiver --query.lookback-delta=2m   
Restart=on-failure  
SuccessExitStatus=0  
LimitNOFILE=65536  
StandardOutput=syslog  
StandardError=syslog  
SyslogIdentifier=prometheus  
​  
[Install]  
WantedBy=multi-user.target  
EOF  
​  
# 启动服务  
systemctl daemon-reload  
systemctl enable prometheus  
systemctl restart prometheus  
systemctl status prometheus

如果想要使用其他时序数据库，如 VM，可以查看文档：[VictoryMetrics](http://n9e.flashcat.cloud/docs/install/victoria/)

## 安装 N9E

cd /opt/package/  
wget https://github.com/ccfos/nightingale/releases/download/v5.14.3/n9e-v5.14.3-linux-amd64.tar.gz  
mkdir /opt/service/n9e  
tar -zxf n9e-v5.14.3-linux-amd64.tar.gz -C /opt/service/n9e

## 导入基础数据

cd /opt/service/n9e/  
mysql -uroot -p1234 < docker/initsql/a-n9e.sql

## 配置启动文件并启动

# N9E Webapi  
cat >/etc/systemd/system/n9e-webapi.service << EOF  
[Unit]  
Description="N9E Webapi"  
After=network.target  
​  
[Service]  
Type=simple  
WorkingDirectory=/opt/service/n9e/  
ExecStart=/opt/service/n9e/n9e webapi -c /opt/service/n9e/etc/webapi.conf  
Restart=on-failure  
SuccessExitStatus=0  
LimitNOFILE=65536  
​  
[Install]  
WantedBy=multi-user.target  
EOF  
​  
systemctl daemon-reload  
systemctl enable n9e-webapi  
systemctl restart n9e-webapi  
systemctl status n9e-webapi  
​  
# N9E Server  
cat >/etc/systemd/system/n9e-server.service << EOF  
[Unit]  
Description="N9E Server"  
After=network.target  
​  
[Service]  
Type=simple  
WorkingDirectory=/opt/service/n9e/  
ExecStart=/opt/service/n9e/n9e server -c /opt/service/n9e/etc/server.conf  
Restart=on-failure  
SuccessExitStatus=0  
LimitNOFILE=65536  
​  
[Install]  
WantedBy=multi-user.target  
EOF  
​  
systemctl daemon-reload  
systemctl enable n9e-server  
systemctl restart n9e-server  
systemctl status n9e-server

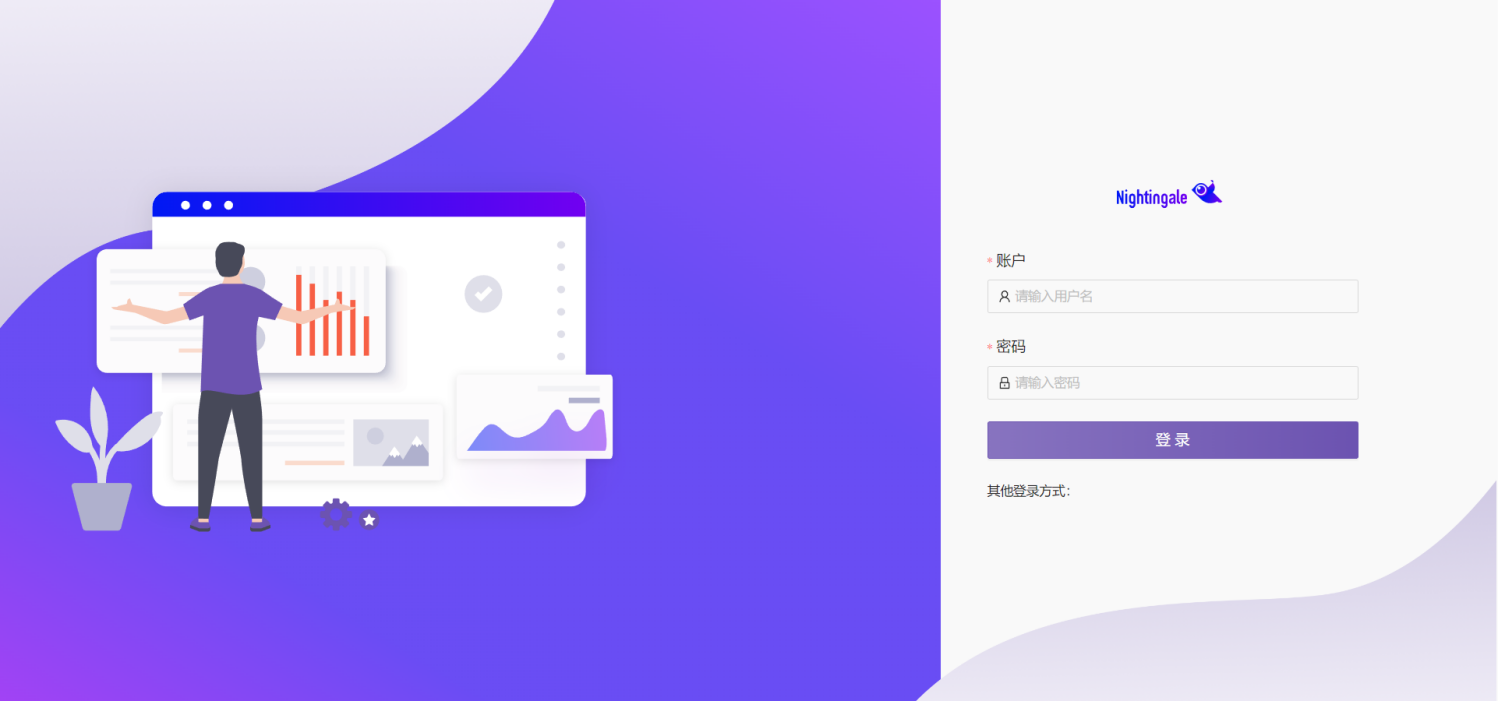
分别监听端口：18000 和 19000

## 安装 Nginx 代理访问

yum -y install nginx  
​  
# 增加反向代理  
server {  
   listen       80;  
   server\_name n9e.ezops.com;  
​  
   location / {  
       proxy\_pass http://127.0.0.1:18000;  
   }  
}  
​  
# 启动 nginx  
nginx

访问查看：<http://113.92.35.218:20000/>

如图所示：默认账号密码：root / root.2020

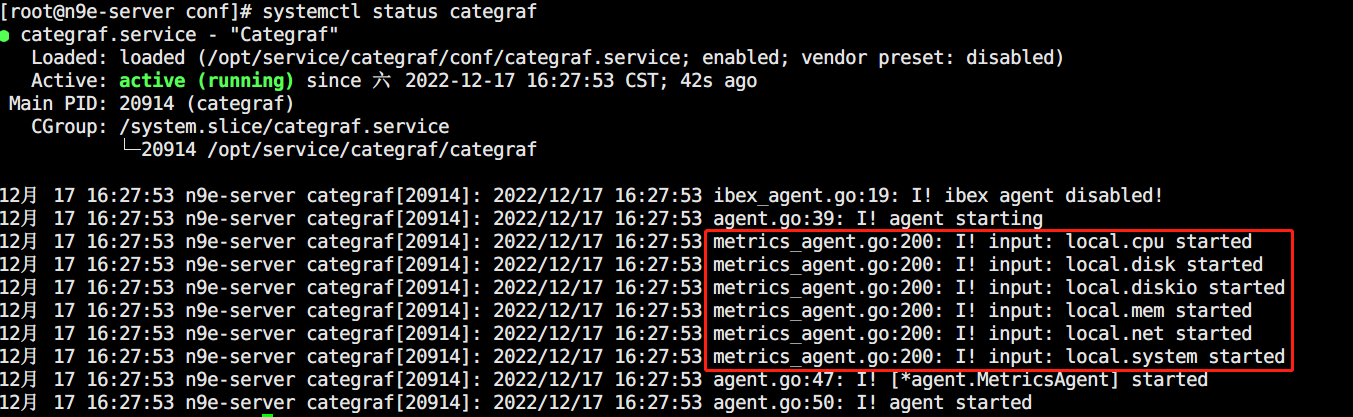


# 八、安装部署采集器

## Categraf

# 解压安装  
cd /opt/package/  
wget https://github.com/flashcatcloud/categraf/releases/download/v0.2.24/categraf-v0.2.24-linux-amd64.tar.gz  
tar -zxf categraf-v0.2.24-linux-amd64.tar.gz   
mv categraf-v0.2.24-linux-amd64 /opt/service/categraf  
​  
# 将 input 开头的目录都先移走  
cd /opt/service/categraf/conf/  
mkdir backup  
mv input.\* backup/  
​  
# 选取需要采集的项目再移动回来相关目录  
cd backup/  
mv input.cpu input.disk input.diskio input.mem input.net input.system ../  
cd ..  
​  
# 修改主配置文件  
vim config.toml  
​  
# 主要修改  
[global]  
# 采集时间间隔  
interval = 15  
# N9E Server 的接口地址  
[[writers]]  
url = "http://192.168.2.210:19000/prometheus/v1/write"  
​  
# 修改启动文件，主要调整里面的目录配置  
vim categraf.service  
​  
# 修改目录  
[Service]  
Type=simple  
ExecStart=/opt/service/categraf/categraf  
WorkingDirectory=/opt/service/categraf  
​  
# 配置软连接然后启动  
ln -s /opt/service/categraf/conf/categraf.service /etc/systemd/system/categraf.service  
systemctl start categraf  
systemctl status categraf  
systemctl enable categraf

此时可以看到启用的模块：



## Telegraf

<https://n9e.github.io/docs/agent/telegraf/>

# 九、平台介绍

1. 对象列表：做业务分组，新建业务组，实现数据查看。
2. 监控看图：
   1. 即时查询：移植的 Prometheus Web UI 查询功能。
      1. 用户自定义监控项
   2. 快捷试图：查看具体主机采集到的数据信息。
   3. 监控大盘：生成类似 Grafana 那样的监控大盘，可以直接导入现有的。
3. 告警管理：
   1. 告警规则配置
   2. 告警屏蔽
4. 人员组织
   1. 用户：配置告警接收方式
   2. 用户 - 团队 - 业务组的关系

# 十、告警自定义

1. 告警通知（自定义通知）：<https://n9e.github.io/docs/usage/notify/>
2. 告警模板：<https://n9e.github.io/docs/usage/format/>