**工业产线数据采集与管理系统**

**1-项目部署文档**

**1.0-2021.3.7**

分工说明

|  |  |
| --- | --- |
| 姓名 | 本文档中主要承担的工作内容 |
| 叶静波 | 后端开发、数据上云 |
| 刘子航 | 前端开发、3D建模 |

版本变更历史

|  |  |  |
| --- | --- | --- |
| 版本 | 提交日期 | 版本说明 |
| 1.0 | 2021.3.7 | 给出基本需求，确定开发环境，部署开发环境和服务器 |

目录

[1. 业务需求 1](#_Toc32562)

[3.2.3 系统整体架构设计 3](#_Toc4779)

[2. 运行与开发环境部署 3](#_Toc12468)

[2.1 运行环境 3](#_Toc3498)

[2.2 开发环境部署 4](#_Toc20553)

[3. 数据需求 9](#_Toc16346)

[3.1 存储需求 9](#_Toc16138)

[3.2 安全传输需求 10](#_Toc15170)

# 业务需求

整个工业产线云平台的技术架构如下图 3.1，本论文主要负责边缘端部分技术，包括消息中间件和存储组件、端设备与边缘设备的数据采集技术、边缘端向微服务或前端提供数据可视化展示技术、边缘端向私有云服务器提供数据上云技术。

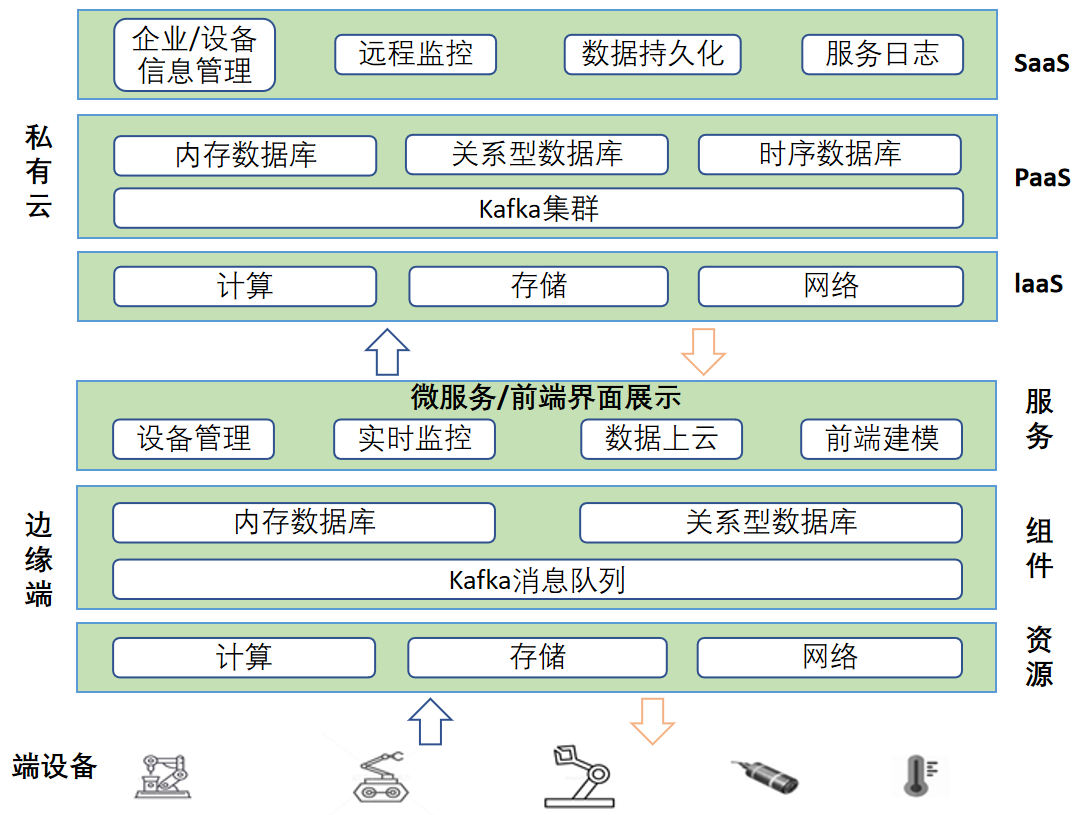


图 1 工业产线云平台技术架构图

工业产线云平台的总体目标是：面向工业产线机器人智能运维需求，利用工业产线中，机器人因流水线工作而产生产生的实时动态数据和环境数据，突破多源时序数据的序列耦合、关联匹配和模式识别等关键技术，建立实时监控和远程指标监控模型（也包括故障诊断和预测性维护），开发工业产线机器人智能运维云服务[2]。如**错误！未定义书签。**，展示了三个方面的服务：基于云平台的工业产线远程监控（包括持久化、上云、前端建模和实时监控）、基于云平台的工业产线状态分析（包括时序数据分析、状态分析）、基于云平台的工业产线预测性维护（包括生产效率分析、使用寿命预测）。

本系统主要功能是：面向工业产线机器人智能运维需求，为实时监控和远程监控与分析提供数据服务，将端设备、机器人和传感器等设备状态，机器人轴电流、位置等信息传输给前端和云服务器。具体如图 3.3显示了数据采集（包括采集数据放入缓冲队列、预处理、数据格式化、持久化操作）和数据管理功能（数据存储、缓存到内存、传输给前端、数据上云）。

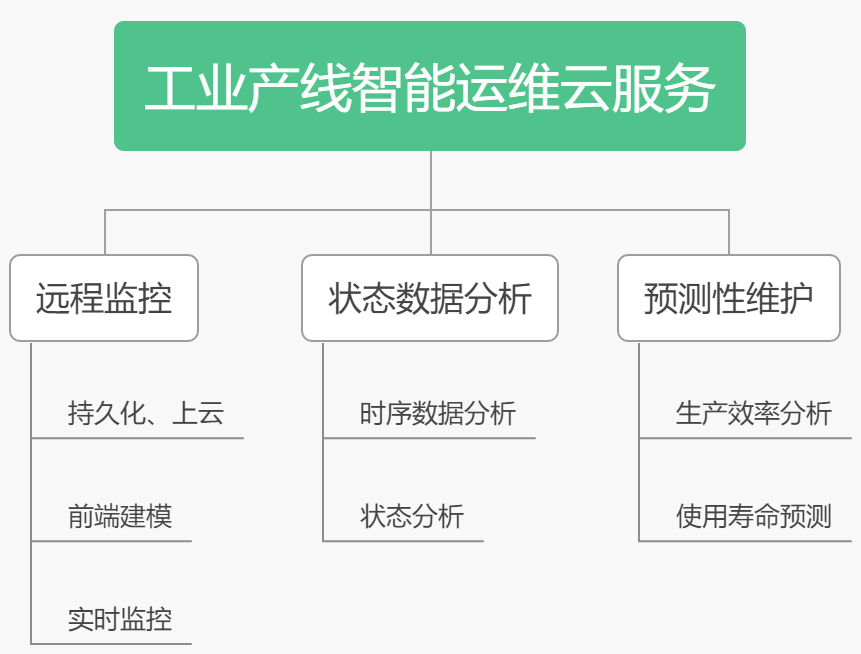


图 3.2 工业产线智能运维服务图

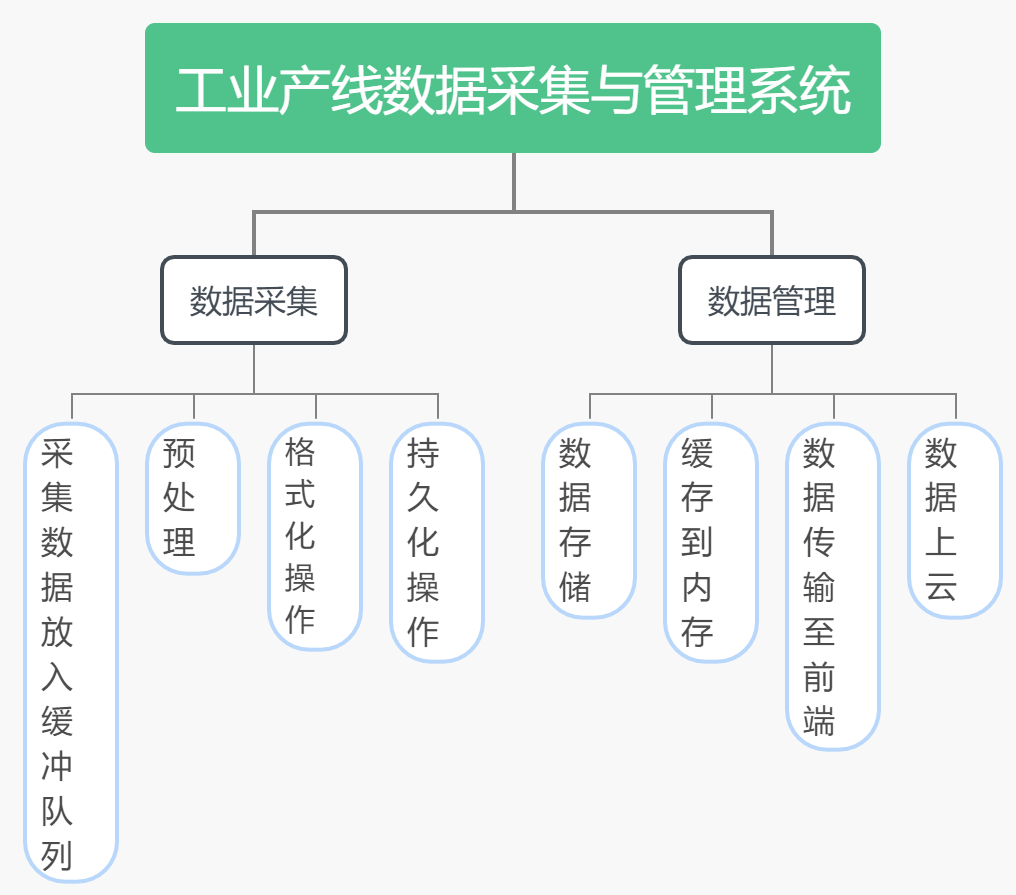


图 3.3 系统主要功能图

### 3.2.3 系统整体架构设计

根据本文第2章系统架构设计综述可以设计出整体的架构图，如图 3.4，红色箭头代表数据采集，绿色箭头代表数据管理，最左边虚框是端设备，中间的虚框是边缘设备，最右边的虚框是云端设备，图中具体展示了所使用的中间件和存储器，以及数据流向。

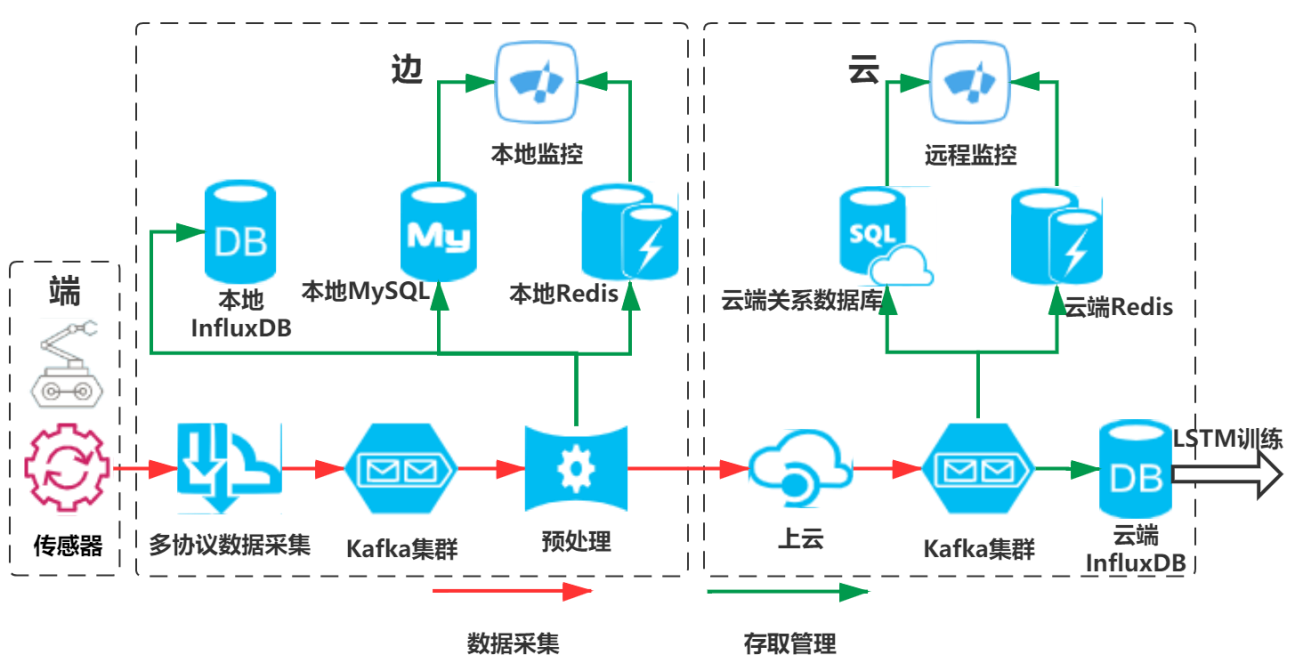


图 3.4 后端整体架构图

# 运行与开发环境部署

## 运行环境

**硬件环境：**

* 机器人生产线（包含机械臂、传感器和数据传输线）

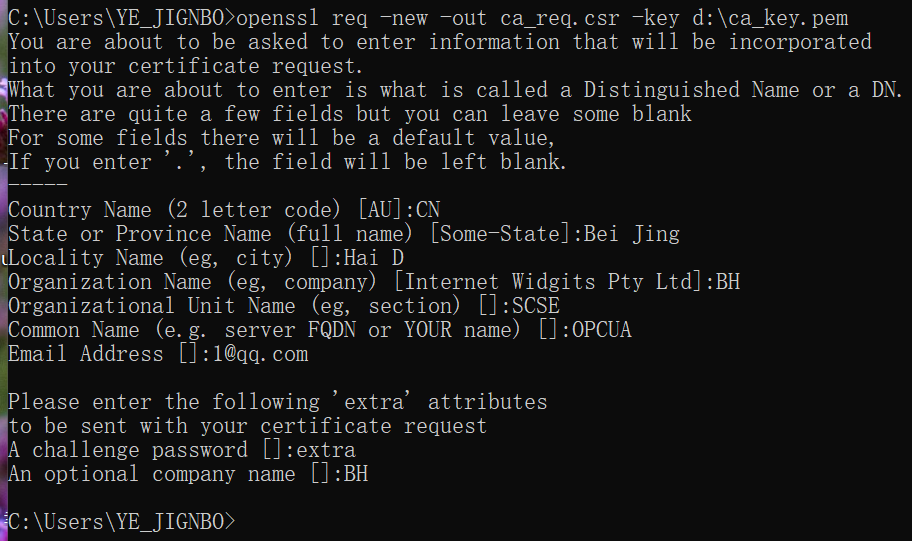
**软件环境：**

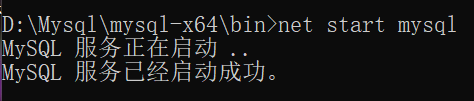
* Windows 10
* Python，node，Django
* 杭州私有云服务器

## 开发环境部署

* Python-3.7、PyCharm Community-2018安装(也可使用专业版)
* cmd状态下Django-3框架安装
* 使用pip 安装django：pip install django；
* 创建Django project：django-admin startproject myDjango；
* 使用PyCharm打开myDjango，在project中设置解释器，安装Django。
* 创建APP：python manage.py startapp workshop，把该文件目录加入到settings文件中的installed\_apps列表里
* admin后台帐号密码：python manage.py createsuperuser； root/123456邮箱：1@qq.com

启动服务：在cmd命令行下输入：python manage.py runserver 0.0.0.0:8000

* OPC UA客户端及OpenSSL部署
* 安装opcua包，安装OpenSSL
* 生成CA密钥：openssl genrsa -out ca\_key.pem 1024
* 创建CA证书：openssl req -new -out ca\_req.csr -key ca\_key.pem
* 生成自签名：openssl x509 -req -in ca\_req.csr -out ca\_cert.pem -signkey ca\_key.pem -days 365
* 再根据CA分别生成服务器和客户端证书、私钥openssl x509 -req -in server\_req.csr -out server\_cert.pem -signkey server\_key.pem -CA ca\_cert.pem -CAkey ca\_key.pem -CAcreateserial -days 365
* 
* MySQL-8.0.23-winx64部署，workbench-8.0可视化工具安装
* 解压缩包
* 配置环境变量，变量名：MYSQL\_HOME；变量值：D:\mysql-x64；
* cd到目录D:\mysql-x64\bin下，执行mysqld --initialize-insecure --user=mysql在上一目录下生成data目录，如果是在PowerShell需要在命令前加“.\”；
* 安装：mysqld -install；（如果denied，则启动管理员运行）
* 启动服务：net start mysql



* 登录MySQL：mysql -u root -p （默认没设置密码，直接回车）
* 设置（或修改）root用户密码：

mysql> use mysql;  
mysql>alter user 'root' @'localhost' identified with mysql\_native\_password by '123456';

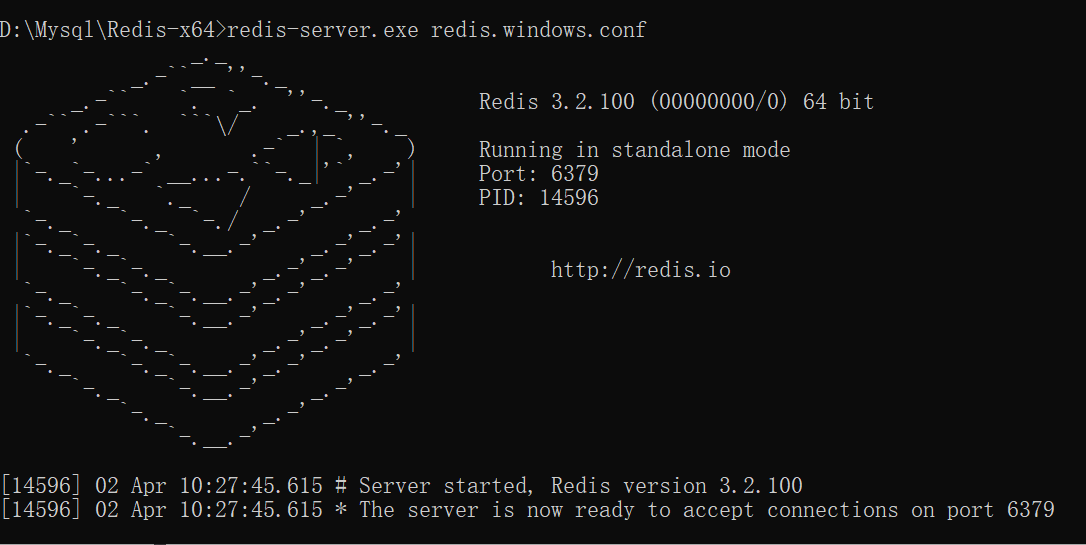
* 退出：mysql> quit

用户名/密码：root/123456

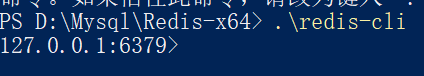
默认端口：127.0.0.1:3306

数据库名称：local\_db

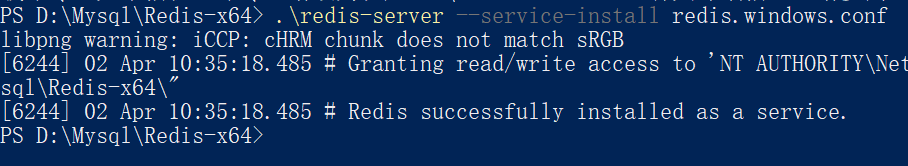
* Redis-5.0部署，可视化工具安装
* 解压，在目录下执行启动服务：redis-server.exe redis.windows.conf 启动后不要关闭cmd，保持启动状态



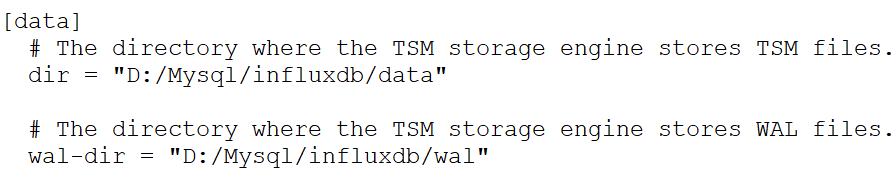
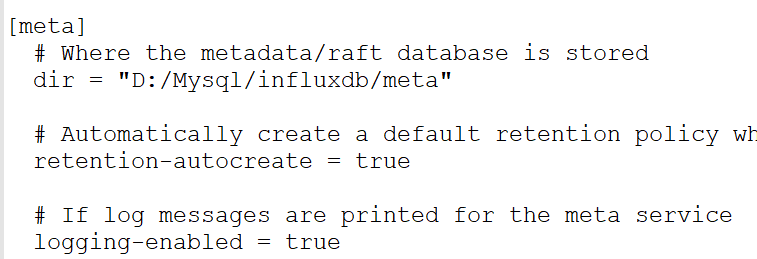
* 新打开cmd窗口，执行：redis-cli 连接默认本机端口号127.0.0.1:6379

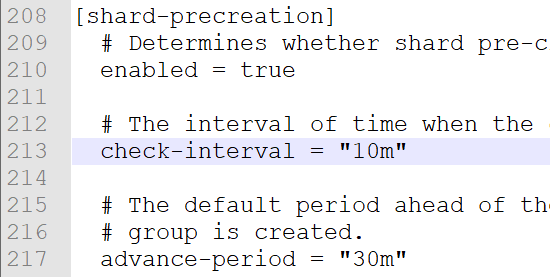
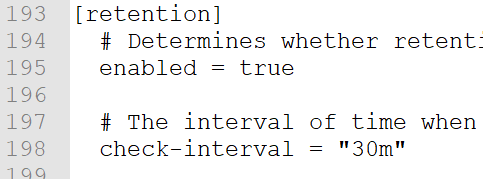


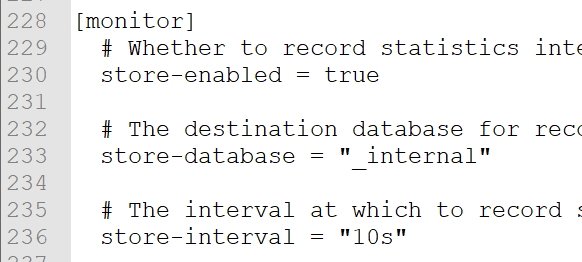
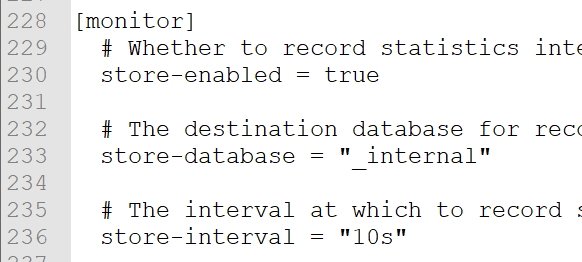
* 将Redis 加入 Windows 服务，执行：redis-server --service-install redis.windows.conf

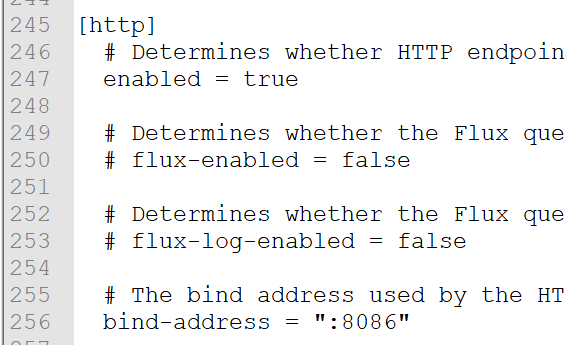


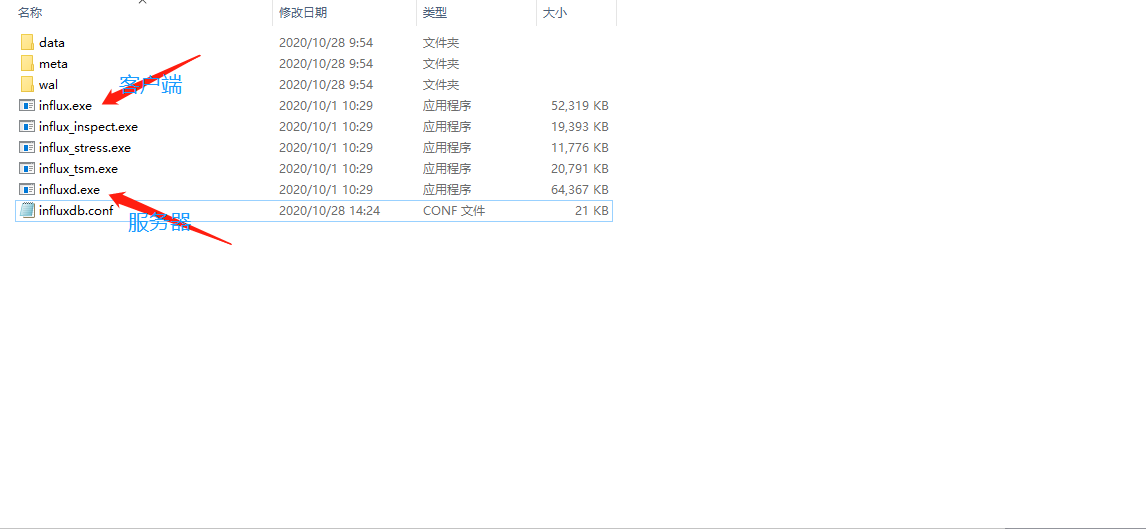
* 密码：123456取消了密码
* 默认端口：127.0.0.1:6379
* auth ‘123456’进入
* Golang，influxDB部署，studio可视化工具安装
* 输入网址：(需要开插件)
* <https://dl.influxdata.com/influxdb/releases/influxdb-1.8.3_windows_amd64.zip>
* 解压，打开配置文件influxdb.conf，修改位置为当前目录D:/Mysql/influxdb，并且删掉相应的注释”#”，然后命令启动influxd.exe -config influxdb.conf生成如下3个文件夹











* 绑定的默认端口：RPC~~127.0.0.1:8088~~
* 启动服务端，客户端连接到http://localhost:8086
* 下载安装Chronograf-1.8.10进行web端实时监控和管理：

[https://dl.influxdata.com/chronograf/releases/chronograf-1.8.10\_windows\_amd64.zip](https://dl.influxdata.com/chronograf/releases/chronograf-1.8.10_windows_amd64.zip" \t "_blank)

* 运行chronograf.exe，默认访问端口[http://localhost:8888](http://localhost:8888/" \t "https://www.cnblogs.com/zouhao/p/_blank)
* 下载解压后直接运行InfluxDBStudio.exe
* 连接influxDB
* 调整时差，需要添加一个golang库
* Kafka部署
* pip安装即可，安装java环境，kafka解压缩包后修改config目录，修改service.properties里面log.dirs路径未log.dirs=E:\\Kafka\\kafka\_2.12-2.0.0\\kafka-logs(注意：文件夹分割符一定要是”\\”)
* 修改zookeeper.properties里面dataDir路径为dataDir=E:\\Kafka\\kafka\_2.12-2.0.0\\data
* 启动kafka内置的zookeeper: zookeeper-server-start.bat ..\..\config\zookeeper.properties，不要关闭窗口，再打开一个
* 在安装目录bin\windows目录下启动kafka服务：kafka-server-start.bat ..\..\config\server.properties
* 端口号zookeeper：2181；kafka：9092
* 如果出现ERROR Shutdown broker错误则删除log文件，重新启动
* 阿里云数据库部署

Kafka集群

# 数据需求

## 存储需求

使用消息队列接收边缘端数据的传输，然后持久化到本地磁盘，同时将热点数据保存到内存数据库

kafka一天的时间过期，Redis取出数据后就删除

修改 server.properties文件中的log.retention.hours=24

数据预处理，需要把机器原始数据转化为角度，以及位姿坐标。然后转化为符合数据库行数据的格式后存储

图片的具体数据通过request.read读取

最终上传到云端，云端自己有一个influxDB存储数据，留给后续的训练

## 安全传输需求

从边缘端采集传输到消息队列的过程采用OPC UA、webSocket、MQTT协议，客户端连接服务器进行传输，进行证书认证和加密传输，保证了数据的安全性。

OPC UA协议规范里建立了安全通道，在客户端中，与服务器建立连接时会先建立安全通道open\_secure\_channel()，之后再根据用户名、密码和认证密钥建立会话。