1. 需求说明

在算法模块，完成异常检测的能力构建。当前仅后端任务，用python实现。
算法模块以指标的形式输出模型计算后的预测值和异常值。

2. 概要设计

2.1.相关概念说明

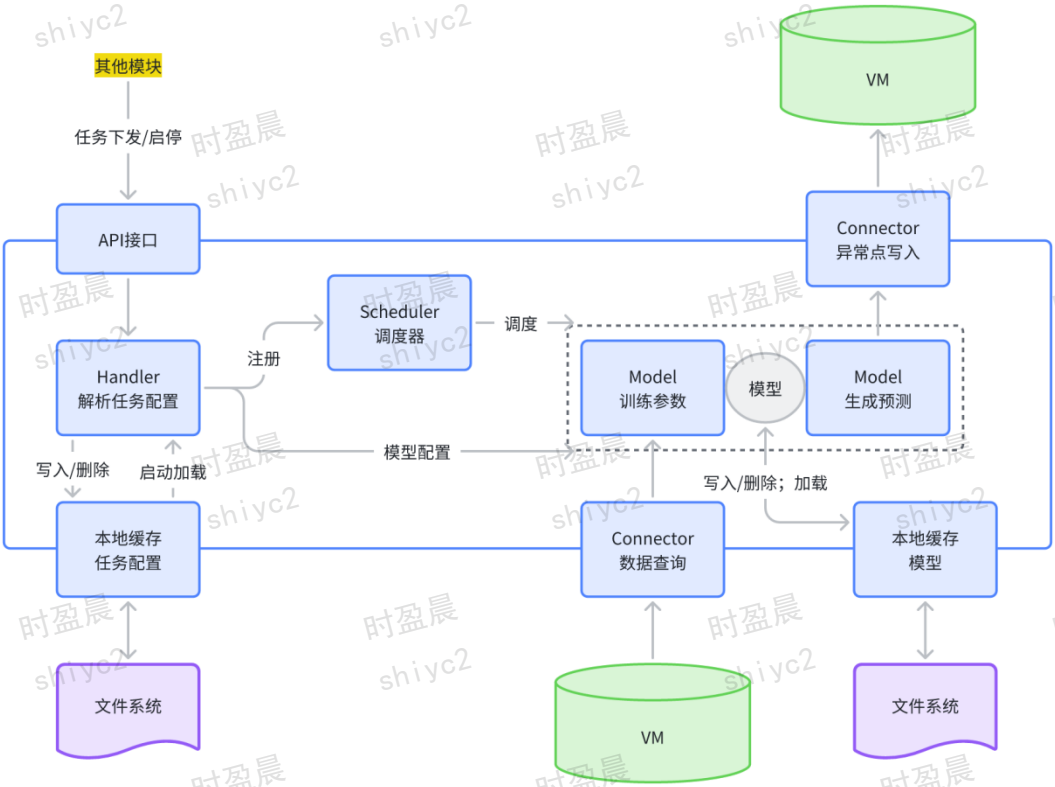
2.1.1. **异常检测**
根据时序指标的历史值建模，分析当前值是否异常
2.1.2. **Handler**
API处理服务，负责解析api接口的指令和配置数据。
2.1.3. **Scheduler**
调度器，负责调度模型参数训练任务和当前异常检测任务。
2.1.4. **Connector**
连接器，负责数据的查询和写入。
2.1.5. **Model**
模型。负责训练和预测。

2.2.流程图

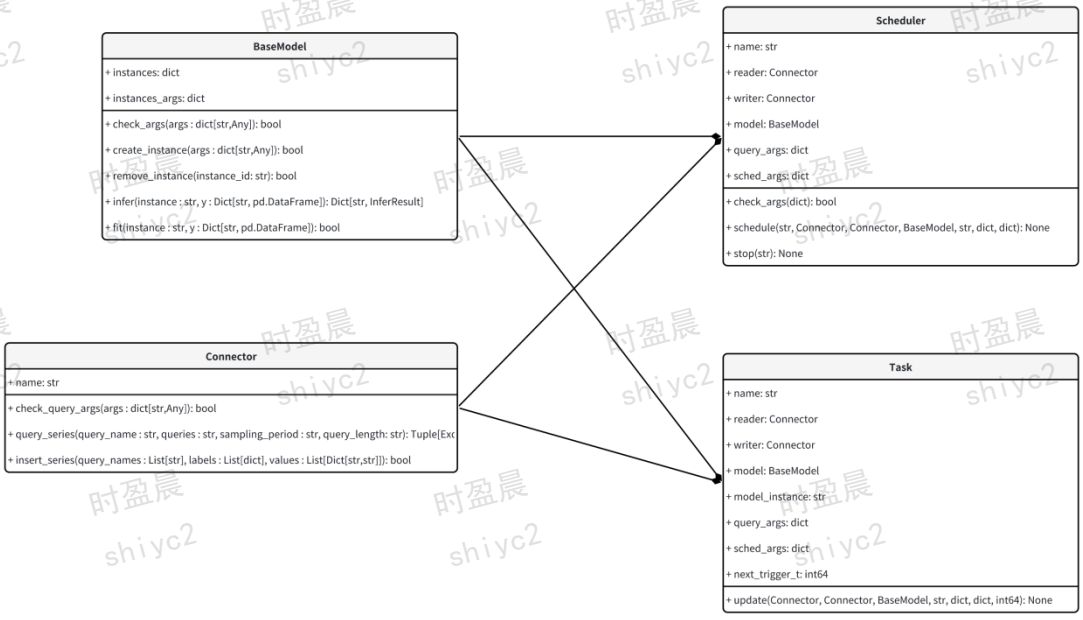
算法模块分为两个能力：第一是预测能力，第二是异常检测能力。

预测能力指的是，采用历史的指标数据训练一个模型（注意，每个时间序列都会对应一个模型），再利用训练好的模型，预测某一段时间内的指标值。

异常检测能力指的是，根据预测的指标值，包括预测的上下限，和真实的指标值进行计算，反应当前真实指标值偏离预测的程度，这个程度即为“异常分数”。


其他模块
任务下发/启停
API接口
Handler
解析任务配置
注册
Scheduler
调度器
调度
模型配置
写入/删除
启动加载
本地缓存
任务配置
文件系统
VM
Connector
异常点写入
Model
训练参数
模型
Model
生成预测
写入/删除；加载
Connector
数据查询
本地缓存
模型
VM
文件系统

2.3.主要类图

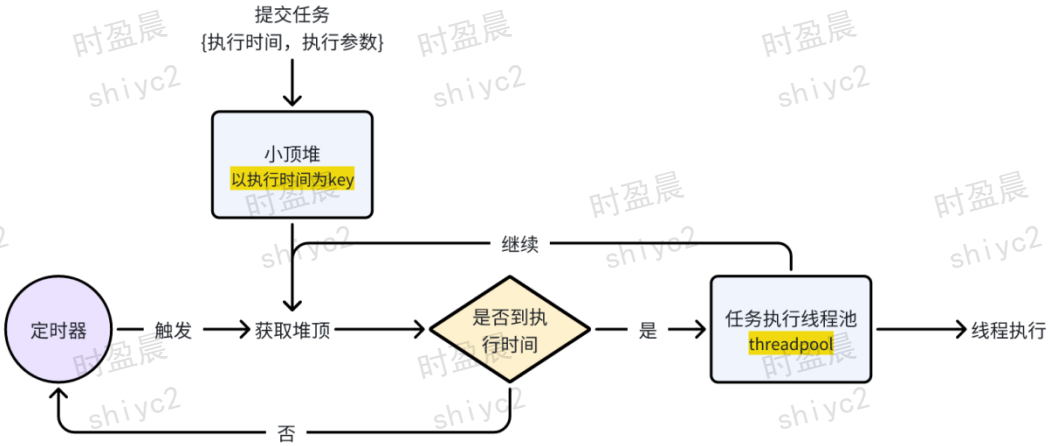

BaseModel
+ instances: dict
+ instances_args: dict
+ check_args(args : dict[str,Any]): bool
+ create_instance(args : dict[str,Any]): bool
+ remove_instance(instance_id: str): bool
+ infer(instance : str, y : Dict[str, pd.DataFrame]): Dict[str, InferResult]
+ fit(instance : str, y : Dict[str, pd.DataFrame]): bool
Scheduler
+ name: str
+ reader: Connector
+ writer: Connector
+ model: BaseModel
+ query_args: dict
+ sched_args: dict
+ check_args(dict): bool
+ schedule(str, Connector, Connector, BaseModel, str, dict, dict): None
+ stop(str): None
Connector
+ name: str
+ check_query_args(args : dict[str,Any]): bool
+ query_series(query_name : str, queries : str, sampling_period : str, query_length: str): Tuple[Exc
+ insert_series(query_names : List[str], labels : List[dict], values : List[Dict[str,str]]): bool
Task
+ name: str
+ reader: Connector
+ writer: Connector
+ model: BaseModel
+ model_instance: str
+ query_args: dict
+ sched_args: dict
+ next_trigger_t: int64
+ update(Connector, Connector, BaseModel, str, dict, dict, int64): None

2.4.组件

2.4.1.调度器

- 周期调度器（Periodic）
  - 调度器参数：

```
单任务参数：
{
            "infer_every":      "多长时间推理一次，必填，格式类似于5s/5m/5h/5d/5w",
             "infer_window":      "推理的时间跨度，必填，格式类似于5s/5m/5h/5d/5w",
              "fit_every":      "多长时间训练一次，必填，格式类似于5s/5m/5h/5d/5w",
            "fit_window":      "训练数据的时间跨度，必填，格式类似于5s/5m/5h/5d/5w"
}
全局参数（在服务的全局配置文件中，提交任务时不用给）：
{
    "max_tasks": -1  //最大任务数，超过则拒绝，-1为不限制
}
```



- 单次任务调度器

  执行一次“训练”+“推理”的步骤。

  单次任务调度器是服务强制加载的，无需启动时指定。
  - 调度器参数

```
{
             "infer_window":      "推理的时间跨度，必填，格式类似于5s/5m/5h/5d/5w",
            "fit_window":      "训练数据的时间跨度，必填，格式类似于5s/5m/5h/5d/5w"
}
```

2.4.2.模型

主要负责指标的异常检测，滚动训练，定时输出指标值的异常分数，分数在(-1,1)可认为此指标在此时刻正常，否则异常，绝对值越高越严重。正数意味着因偏高而异常，负数意味着因偏低而异常。

当前支持以下模型：

1. Prophet模型（Prophet）
   - 模型参数（所有参数均**选填**，具体含义参考官网https://facebook.github.io/prophet/docs/quick_start.html#python-api）

```
{
    "growth": "linear",
    "changepoints": null,
    "n_changepoints": 25,
    "changepoint_range": 0.8,
    "yearly_seasonality": "auto",
    "weekly_seasonality": "auto",
    "daily_seasonality": "auto",
    "holidays": null,
    "seasonality_mode": "additive",
    "seasonality_prior_scale": 10,
    "holidays_prior_scale": 10,
    "changepoint_prior_scale": 0.05,
    "mcmc_samples": 0,
    "interval_width": 0.8,
    "uncertainty_samples": 1000
}
```

2.4.3.输入输出

- 支持的连接器列表：
  - Victoriametrics：输入、输出
  - Pulsar：输出
  - 标准输入输出：输出
- 输出指标：查询表达式返回的label都会添加到输出指标中，任务参数中的${query_name}会写到标签“__for”中。
  - _model_output_y：指标原始值，仅异常检测任务有此指标
  - _model_output_yhat：指标预测中位值
  - _model_output_yhat_upper：指标预测范围的上限
  - _model_output_yhat_lower：指标预测范围的下限
  - _model_output_anomaly_score：指标异常分数，仅异常检测任务有此指标

3. 数据库设计

无

4. 接口设计

– 接口列表

- 任务提交

【注】若提交时name重复，则会更新此任务。更新任务为CPU高负载工作，故尽量不要重复提交任务。

URL	srv-algorithm:8450/submit/{tenant}
参数	tenant：租户ID，int64，是数据源的租户
调用方法	POST
请求体	{ "name": "task名称，自定义，特殊字符仅支持'-''_''.'，每个租户唯一，必填", "reader": "数据查询组件，选择，范围由服务配置文件确定（配置文件中connector.name，pipline包含reader的）", "writer": "数据写入组件，选择，范围由服务配置文件确定（配置文件中connector.name，pipline包含writer的）", "model": "模型，选择，范围由服务配置文件确定（配置文件中model.name）", "model_args": {}, // 模型参数，由具体模型确定 "scheduler": "调度器，选择，范围由服务配置文件确定（配置文件中scheduler.name）", "scheduler_args": {}, // 调度器参数，有具体调度器确定 "query_name": "本次任务的查询名称，用于输出指标的_for标签，每个租户唯一", "query_args": { "queries": "查询表达式", "sampling_period_fit": "训练采样间隔，例如：5m", "sampling_period_infer": "推理采样间隔，例如：1m" } }
响应体	失败返回错误信息； 成功返回'success'

- 任务停止

URL	srv-algorithm:8450/stop/{tenant}
参数	tenant：租户ID，int64，是数据源的租户
调用方法	POST

请求体	{
	"schedule": "调度器，选择，范围由服务配置文件确定",
	"name": "任务名称"
	}

响应体	失败返回错误信息； 成功返回'success'

- 模型测试

执行单次训练和检测，返回结果

URL	srv-algorithm:8450/test/{tenant}
参数	tenant：租户ID，int64，是数据源的租户
调用方法	POST

请求体	{
	"reader": "数据查询组件，选择，范围由服务配置文件确定（配置文件中connector.name，pipline包含reader的）",
	"writer": "数据写入组件，选择，范围由服务配置文件确定（配置文件中connector.name，pipline包含writer的）",
	"model": "模型，选择，范围由服务配置文件确定（配置文件中model.name）",
	"model_args": {}, // 模型参数，由具体模型确定
	"scheduler_args": { // 单次任务调度器参数
	"infer_window": "12h",
	"fit_window": "12h"
	},
	"query_name": "本次任务的查询名称",
	"query_args": {
	"queries": "查询表达式",
	"sampling_period_fit": "训练采样间隔，例如：5m",
	"sampling_period_infer": "推理采样间隔，例如：1m"
	}
	}

响应体	{
	"anomaly": { // 异常检测结果
	"cpu_usage-202.4": { // query_name
	"data": { // 和prometheus query_range结构一致
	"result": [
	{
	"metric": {
	"__name__": "_model_output_y",
	"_res_id": "507215006502145",
	"_res_ip": "192.168.202.4",
	"_res_model": "Switch",
	"_task_id": "1868911926194339841"
	},
	"values": [
	[
	1734483840.0,
	"6.0"

```
                        ],
                        ...
                    ]
                },
                {...}
            ]
        }
    },
    "original": {       // 原始查询结果
        "cpu_usage-202.4": { // query_name
            "data": {  // 和prometheus query_range结构一致
                "result": [...]
            }
        }
}
```

– 任务配置示例

每个任务在服务的本地文件系统都存有一份配置信息，路径在${home}/task/${tenantid}/${task-name}.yaml，其中${home}是服务数据目录的根路径，${tenantid}是租户ID（需要和数据源，例如VM使用的租户ID保持一致），${task-name}是task名称（即），每个租户下唯一。一个推荐配置如下：

```yaml
model_args: {}
model_name: prophet
name: yc-task
query_args:
  queries: os.mem.used
  sampling_period_fit: 5m
  sampling_period_infer: 1m
query_name: os_mem_used
reader_name: vm-source
scheduler_args:
  fit_every: 1d
  fit_window: 14d
  infer_every: 1m
  infer_window: 5m
scheduler_name: periodical
writer_name: vm-source
```

配置中的字段与提交任务接口中的字段一一对应。

5. 部署说明

服务由python编写，需要的基础环境为python3.10。安装其他依赖可以执行：

```
pip install -r requirements.txt
```

如果需要部署镜像，本地执行：

```
pip download -r ./requirements.txt -d ./lib/
```

将包下载到lib文件夹下，一并打入基础镜像中，在dockerbuild时执行包安装：

```
pip install --no-index --find-links=./lib -r ./requirements.txt
```

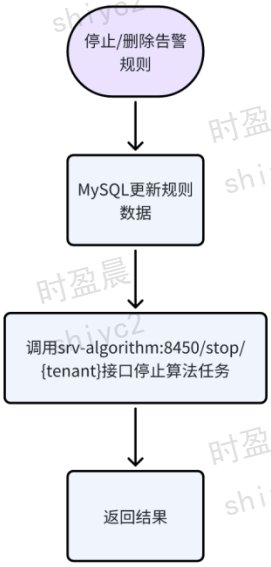
资源限制：视数据量，启动内存占用大概80M，单条时间序列训练的模型可以按照？计算内存占用。

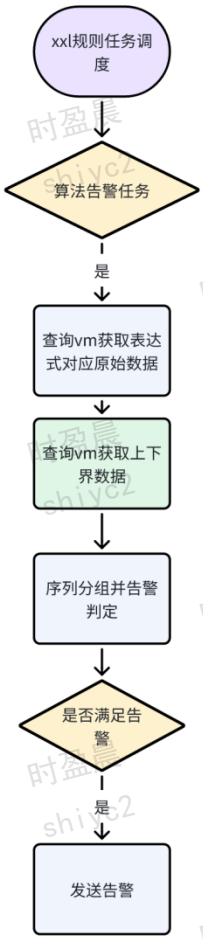
业务指标定义（promethues）—暂未开发

6. 监控器改造点

5.1. 后端

5.1.1. 接口&告警判定逻辑





5.1.2. 算法任务参数

算法任务参数	对应取值
name	告警规则名
reader	默认vm-source

writer	默认vm-source
model	根据告警规则中配置的范型id，取自范型数据
model_args	根据告警规则中配置的范型id，取自范型数据
model_args.interval_width	对应告警规则中配置的置信区间比例值 $.config.intervalWidth
scheduler	默认periodical

scheduler_args	默认：
	{
	"fit_every": "1d",
	"fit_window": "14d",
	"infer_every": "1m",
	"infer_window": "5m"
	}

query_name	detector-rule-{告警规则id}

query_args	"query_args": {
	"queries": "{{查询表达式}}",
	"sampling_period_fit": "5m",
	"sampling_period_infer": "{{execInterval}}"
	}

5.1.3. 接口

5.1.3.1. 算法范型接口

URI	/srv-detector/anomaly-detection/paradigm/list
Method	GET

Response	{
	"code": 0,
	"msg": "成功",
	"data": [
	{
	"id": "asdasasda",
	"name": "",
	"param": {
	"growth": "linear",
	"changepoints": null,
	"n_changepoints": 25,
	"changepoint_range": 0.8,
	"yearly_seasonality": "auto",
	"weekly_seasonality": "auto",
	"daily_seasonality": "auto",
	"holidays": null,
	"seasonality_mode": "additive",
	"seasonality_prior_scale": 10,
	"holidays_prior_scale": 10,
	"changepoint_prior_scale": 0.05,
	"mcmc_samples": 0,
	"interval_width": 0.8,
	"uncertainty_samples": 1000
	}
	}
	]
	}

5.1.3.2. 算法训练结果趋势图

字段	类型	描述
$.fitWindow	String	训练数据的时间跨度，必填，格式类似于5s/5m/5h/5d/5w
$.config.type	String	规则类型，此处算法异常检测type值为ANOMALY_DETECTION
$.config.algorithmParadigmId	String	算法范型id，值为7.1.3.1算法范型接口的列表元素id（$.data[*].id）
$.config.intervalWidth	double	控制置信区间的宽度，即算法结果上下界宽度，默认0.8
$.config.compareType	String	比较类型：大于上界-GT_UPPER、小于下界-LT_LOWER、大于上界或小于下界-OUT_OF_RANGE

URI	/srv-detector/anomaly-detection/query-range
Method	PSOT
Request	{ "fitWindow": "12h", "config": { "algorithmParadigmId": "common_paradigms", "intervalWidth": 0.8, "compareType": "OUT_OF_RANGE", "queryConfig": { "metricCode": "cpu.usage", "aggreType": "AVG", "groupBys": ["_res_ip", "_res_model"], "selectConditions": [ ], "editorMode": "builder" }, "type": "ANOMALY_DETECTION", "alertCriteria": [{ "level": "CRITICAL", "triggerCount": 1, "logical": "AND", "conditions": [{ "operator": "EQUAL", "value": 1 }] }, { "level": "MAJOR", "triggerCount": 1, "logical": "AND",

	"conditions": [
	{
	"operator": "EQUAL",
	"value": null
	}
	]
	},
	{
	"level": "MODERATE",
	"triggerCount": 1,
	"logical": "AND",
	"conditions": [
	{
	"operator": "EQUAL",
	"value": null
	}
	]
	},
	{
	"level": "MINOR",
	"triggerCount": 1,
	"logical": "AND",
	"conditions": [
	{
	"operator": "EQUAL",
	"value": null
	}
	]
	}
	],
	"noDataTriggerCount": 1,
	"noDataLevel": null,
	"resolveTriggerCount": 1,
	"abnormalDuration": "1"
	},
	"execInterval": "1m"
	}

Response	{
	"msg": "处理成功！",
	"body": [
	[
	{
	"metric": {
	"_res_ip": "192.168.202.3",
	"_res_model": "Switch"
	},
	"values": [
	{
	"1734542940.0": "3.0"
	},
	{
	"1734543240.0": "3.0"
	}
	]
	},
	{

```
                "metric": {
                  "_res_ip": "192.168.202.4",
                  "_res_model": "Switch"
                },
                "values": [
                  {
                    "1734543060.0": "3.0"
                  },
                  {
                    "1734543360.0": "3.0"
                  }
                ]
              },
              {
                "metric": {
                  "_res_ip": "192.168.235.10",
                  "_res_model": "OS"
                },
                "values": [
                  {
                    "1734543000.0": "1.0"
                  },
                  {
                    "1734543300.0": "1.0"
                  }
                ]
              }
            ],
            [
              {
                "metric": {
                  "__name__": "_model_output_y",
                  "_res_ip": "192.168.202.3",
                  "_res_model": "Switch"
                },
                "values": [
                  {
                    "1734542940.0": "3.0"
                  },
                  {
                    "1734543240.0": "3.0"
                  }
                ]
              },
              {
                "metric": {
                  "__name__": "_model_output_yhat",
                  "_res_ip": "192.168.202.3",
                  "_res_model": "Switch"
                },
                "values": [
                  {
                    "1734542940.0": "1.0"
                  },
                  {
                    "1734543240.0": "1.0"
                  }
```

```
                ]
            },
            {
                "metric": {
                    "__name__": "_model_output_yhat_lower",
                    "_res_ip": "192.168.202.3",
                    "_res_model": "Switch"
                },
                "values": [
                    {
                        "1734542940.0": "0.9999999987394873"
                    },
                    {
                        "1734543240.0": "0.9999999986925525"
                    }
                ]
            },
            {
                "metric": {
                    "__name__": "_model_output_yhat_upper",
                    "_res_ip": "192.168.202.3",
                    "_res_model": "Switch"
                },
                "values": [
                    {
                        "1734542940.0": "1.0000000013235644"
                    },
                    {
                        "1734543240.0": "1.0000000013043118"
                    }
                ]
            },
            {
                "metric": {
                    "__name__": "_model_output_anomaly_score",
                    "_res_ip": "192.168.202.3",
                    "_res_model": "Switch"
                },
                "values": [
                    {
                        "1734542940.0": "1547941451.3578053"
                    },
                    {
                        "1734543240.0": "1531534728.9098954"
                    }
                ]
            }
        ]
    ],
    "httpStatusCode": 200
}
```


5.1.3.3. 创建监控器告警规则

字段	类型	描述

$.type	String	规则类型，此处算法异常检测type值为ANOMALY_DETECTION
$.config.type	String	规则类型，此处算法异常检测type值为ANOMALY_DETECTION
$.config.algorithmParadigmId	String	算法范型id，值为7.1.3.1算法范型接口的列表元素id（$.data[*].id）
$.config.intervalWidth	double	控制置信区间的宽度，即算法结果上下界宽度，默认0.8
$.config.compareType	String	比较类型：大于上界-GT_UPPER、小于下界-LT_LOWER、大于上界或小于下界-OUT_OF_RANGE

URI	/srv-monitor/monitor-strategy/add
Method	PSOT

Request	{
	"resType": "PhysicalServer",
	"name": "qweqwe",
	"defaultType": 0,
	"desc": null,
	"resIds": null,
	"resGroupIds": null,
	"rules": [
	{
	"metricCode": "cpu.usage",
	"status": 1,
	"metricName": "CPU利用率",
	"config": {
	"algorithmParadigmId": "asdasdasdas",
	"intervalWidth": 0.8,
	"compareType": "OUT_OF_RANGE",
	"queryConfig": {
	"metricCode": "cpu.usage",
	"aggreType": "AVG",
	"groupBys": [
	"_res_ip",
	"_res_model"
	],
	"selectConditions": [
	{
	"label": "_res_ip",
	"value": "192.168.202.3",
	"logic": "AND",
	"op": "EQUAL"
	}
	],
	"editorMode": "builder"
	},
	"type": "ANOMALY_DETECTION",
	"alertCriteria": [
	{

```
                "level": "CRITICAL",
                "triggerCount": 1,
                "logical": "AND",
                "conditions": [
                  {
                    "operator": "EQUAL",
                    "value": 1
                  }
                ]
              },
              {
                "level": "MAJOR",
                "triggerCount": 1,
                "logical": "AND",
                "conditions": [
                  {
                    "operator": "EQUAL",
                    "value": null
                  }
                ]
              },
              {
                "level": "MODERATE",
                "triggerCount": 1,
                "logical": "AND",
                "conditions": [
                  {
                    "operator": "EQUAL",
                    "value": null
                  }
                ]
              },
              {
                "level": "MINOR",
                "triggerCount": 1,
                "logical": "AND",
                "conditions": [
                  {
                    "operator": "EQUAL",
                    "value": null
                  }
                ]
              }
            ],
            "noDataTriggerCount": 1,
            "noDataLevel": null,
            "resolveTriggerCount": 1,
            "abnormalDuration": "1"
          },
          "unit": "%",
          "metricType": 2,
          "description": "CPU利用率是指在一定时间内，CPU执行指令的比率。它通常用来衡量系统计算资源的使用情况",
          "execInterval": "1m",
          "alertTemplateConfig": {
            "type": 0,
```

	"alert": "${resource.typeName}(名称：${resource.name}，IP：${resource.ip})，${query.metricName}触发【${rule.level}】告警，最近值为${rule.result}。",
	"resolve": "${resource.typeName}(名称：${resource.name}，IP：${resource.ip})，${query.metricName}告警【${rule.level}】，最近值为${rule.result}。",
	"noData": "${resource.typeName}(名称：${resource.name}，IP：${resource.ip})，${query.metricName}的最近数据为空，触发【${rule.level}】告警。"
	},
	"type": "ANOMALY_DETECTION",
	"delayConfig": {
	
	},
	}
	],
	"monitoringPeriodConfig": [
	{
	"dayOfWeeks": [
	"MONDAY",
	"TUESDAY",
	"WEDNESDAY",
	"THURSDAY",
	"FRIDAY",
	"SATURDAY",
	"SUNDAY"
	],
	"startTime": "00:00",
	"endTime": "23:59"
	}
	],
	"dataType": 0,
	"status": 0
	}

Response	{
	"code": 0,
	"msg": "成功",
	"data": {}
	}

5.1.3.4. 编辑监控器告警规则

URI	/srv-monitor/monitor-strategy/edit
Method	PSOT
Request	{ "resType": "PhysicalServer", "name": "qweqwe", "defaultType": 0, "desc": null, "resIds": null, "resGroupIds": null, "rules": [{ "metricCode": "cpu.usage", "status": 1,

```
"metricName": "CPU利用率",
"config": {
  "algorithmParadigmId": "asdasdasdas",
  "intervalWidth": 0.8,
  "compareType": "OUT_OF_RANGE",
  "queryConfig": {
    "metricCode": "cpu.usage",
    "aggreType": "AVG",
    "groupBys": [
      "_res_ip",
      "_res_model"
    ],
    "selectConditions": [
      {
        "label": "_res_ip",
        "value": "192.168.202.3",
        "logic": "AND",
        "op": "EQUAL"
      }
    ],
    "editorMode": "builder"
  },
  "type": "ANOMALY_DETECTION",
  "alertCriteria": [
    {
      "level": "CRITICAL",
      "triggerCount": 1,
      "logical": "AND",
      "conditions": [
        {
          "operator": "GT",
          "value": 80
        }
      ]
    },
    {
      "level": "MAJOR",
      "triggerCount": 1,
      "logical": "AND",
      "conditions": [
        {
          "operator": "GT",
          "value": null
        }
      ]
    },
    {
      "level": "MODERATE",
      "triggerCount": 1,
      "logical": "AND",
      "conditions": [
        {
          "operator": "GT",
          "value": null
        }
      ]
    },
```

```
                {
                  "level": "MINOR",
                  "triggerCount": 1,
                  "logical": "AND",
                  "conditions": [
                    {
                      "operator": "GT",
                      "value": null
                    }
                  ]
                }
              ],
              "noDataTriggerCount": 1,
              "noDataLevel": null,
              "resolveTriggerCount": 1,
              "abnormalDuration": "1"
            },
            "unit": "%",
            "metricType": 2,
            "description": "CPU利用率是指在一定时间内，CPU执行指令的比率。它通常用来衡量系统计算资源的使用情况",
            "execInterval": "1m",
            "alertTemplateConfig": {
              "type": 0,
              "alert": "${resource.typeName}(名称：${resource.name}，IP：${resource.ip})，${query.metricName}触发【${rule.level}】告警，最近值为${rule.result}。",
              "resolve": "${resource.typeName}(名称：${resource.name}，IP：${resource.ip})，${query.metricName}告警【${rule.level}】，最近值为${rule.result}。",
              "noData": "${resource.typeName}(名称：${resource.name}，IP：${resource.ip})，${query.metricName}的最近数据为空，触发【${rule.level}】告警。"
            },
            "type": "ANOMALY_DETECTION",
            "delayConfig": {

            },
            "id": "1869209345426874369"
          }
        ],
        "monitoringPeriodConfig": [
          {
            "dayOfWeeks": [
              "MONDAY",
              "TUESDAY",
              "WEDNESDAY",
              "THURSDAY",
              "FRIDAY",
              "SATURDAY",
              "SUNDAY"
            ],
            "startTime": "00:00",
            "endTime": "23:59"
```

	}], "dataType": 0, "id": "1869209345362460674", "status": 0 }

Response	{
	"code": 0,
	"msg": "成功",
	"data": {}
	}

5.2. 前端

5.2.1. 指标趋势图

如果当前告警规则类型为算法异常检测时，指标趋势图接口除了根据表达式查询指标，还需要查询

5.2.2. 创建规则相关页面