

## 入口:

new inspection对象: 新建指标数据存放对象、加载配置文件数据、初始化公共变量、线程池、构建输出路径等辅助工作; 开启守护线程

集群bean, 节点bean, 服务bean, 表bean balabal所有对象数据都存在Data类里, 各种指标啥的

GlobalArgs类里存放公共变量

这些辅助性的类都存在common包, bean包存对象数据原型, 最终都会放进Data类, util包存工具类

然后执行run方法。

1. check几大类, Cluster、Node、Service、Hdfs、Table, 各个项目下面还有子项目, 自己ctrl左键看;
2. 然后测试ping、tcp延迟什么的, 不重要;
3. 然后比较check数据分析结果comparisonResult();
4. 然后关闭巡检服务closeInspection()其下outputReport()输出报告, 这个很重要, 主要看check收集信息和outputReport输出报告, 具体三种方式(xls, json, html)都在大的report子包下有对应的接口和实现

根据以上, 所有check项目的包下都有三个子包: comparison、report、servlet, 下面都是些Impl类。servlet下的impl负责具体的收集某一项巡检工作的数据, report下的impl负责在报告里输出哪些内容。

以收集prometheus发出的监控告警信息为例。可以通过 ip:host/api/v1/alerts的方式拿到prometheus监控到的Alerts信息, 需要嵌入到Inspection内并输出巡检报告。

### 1 搜集

- 新增收集prometheus监控信息的代码, 按照inspection现有的结构: 建议嵌在ClusterCheck下, 它做的是集群整体的指标检查。ClusterCheck实现了CheckInterface, 根据具体子项目的不同选择不同的servlet包下的check实现。那么在ClusterCheck类line 66 要加上case MONITORALERT。相应的在OutputItemEnum枚举类型里加上MONITORALERT (要加上对应的JSON和XLS output实现类, todo) 表示输出的项目里有这么个东西, 然后OutputSelectBean构造也得加addOutputItem(OutputItemEnum.MONITORALERT);
- 然后接着完成monitorialert的具体收集类实现, 创建GetInfoOfMonitorAlerts, 应该在cluster.servlet包下, 重写getInfo方法, 通过httpClient巴拉巴拉的拿到JSON对象。
- prometheus的地址不写得死在GlobalArgs line 58加一句: monitorAlertUrl = env.getProperty("monitorAlertUrl",""); env是data.loadProp中加载进来的env.properties文件的对象。所以也得在env.properties中创建一个monitorAlertUrl, 其值可以不写, 可以在inspection.yaml文件里配置默认路径, 届时将在巡检工具的页面下方由使用者指定要使用什么路径。
- 存到哪儿呢? 新建一个MonitorAlertBean对象, setter getter写全, 构造器传入JSON对象, 解析成对应的字符串。MonitorAlertBean在common.bean包下。
- 既然创建了一个MonitorAlertBean, 那也得在Data类里加上monitorAlerts成员, 它是MonitorAlertBean的List容器, 加上addMonitorAlert和getMonitorAlert方法。也就是说通过http restful api拿到的json格式的监控告警, 解析后封装成monitorialertbean, 添加到Data对象的monitorAlerts列表里, 因为不会只有一个告警, 大多数时候有多个。

### 2 输出

```
private void outputReport() {
    Report report = new Report(data);
    String timeStamp = new SimpleDateFormat("yyyy-MM-dd").format(System.currentTimeMillis());
    report.outputXls(timeStamp);
    report.outputJson(timeStamp);
    report.outputHtml(timeStamp);
    report.outputTableError(timeStamp);
    report.outputAiopsData(timeStamp);
}
```

三大类，先做了HTML，json和xls还没具体实现。

```
public void outputHtml(String timestamp) {
    SummaryReport report = new SummaryReport(data);
    report.outputReport(GlobalArgs.goalPath, timestamp);
}
```

summaryreport类，把监控告警加到了集群稳定性检查下面。那就应该在createItem方法下的ClusterSummaryContentImpl实现类里加。

在loadCheckItems下新加一个 line 30 ~ 32

```
if (GlobalArgs.checkSelect.needOutput(OutputItemEnum.MONITORALERT)) {
    super.addCheckItem("监控告警信息", MonitorAlertContentItemImpl.class);}
}
```

再实现具体的MonitorAlertContentItemImpl类，其内容就是html语句，给页面显示排一下版，在report.html.cluster包下。里面的GlobalArgs.monitorUrl是grafana的页面地址，配置和monitorAlertUrl一样。主意当告警触发是pengding时颜色未orange，firing时为红色。