

解决方案实践

# 永洪商业智能解决方案实践

文档版本 1.0  
发布日期 2024-03-01



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

---

# 目录

---

|                        |           |
|------------------------|-----------|
| <b>1 方案概述</b> .....    | <b>1</b>  |
| <b>2 资源和成本规划</b> ..... | <b>4</b>  |
| <b>3 实施步骤</b> .....    | <b>11</b> |
| <b>4 附录</b> .....      | <b>17</b> |
| 4.1 背景信息.....          | 17        |
| 4.2 常见问题.....          | 17        |
| 4.2.1 安装时遇到的问题.....    | 18        |
| 4.2.2 启动时遇到的问题.....    | 20        |
| <b>5 修订记录</b> .....    | <b>24</b> |

# 1 方案概述

## 应用场景

在数字化时代，企业面临用户需求变化、业务流程变化、以及数据量不断增长等一系列挑战，数字化转型已成为企业核心战略，是企业未来业务创新和增长的重要驱动力。在经历了人口红利、城镇化红利和互联网红利后，数字化转型成为企业未来增长的唯一红利。商业智能BI让企业“一把手”快速认知数字化价值的最佳手段之一，在企业数字化建设中扮演了极其重要的角色。

永洪商业智能解决方案为企业提供从数据采集、清洗、整合、存储、计算、建模、训练、展现、协作等全流程数据分析，预测市场走向、优化供应链、支持决策制定，为企业提供了有效的数据分析和决策支持工具，帮助企业更好的了解市场、优化业务、降低风险，更加科学和高效地管理企业，推动企业向更加智能化的方向发展。

## 方案架构

本方案由华为云底座、数据分析平台和数据应用平台组成，基于华为云能力，构建“一个平台、两个统一、三个层级、四个改善”的数据分析系统：

一个平台：搭建统一数据分析和报表管理平台；

两个统一：数据规范统一，报表分析规范统一；

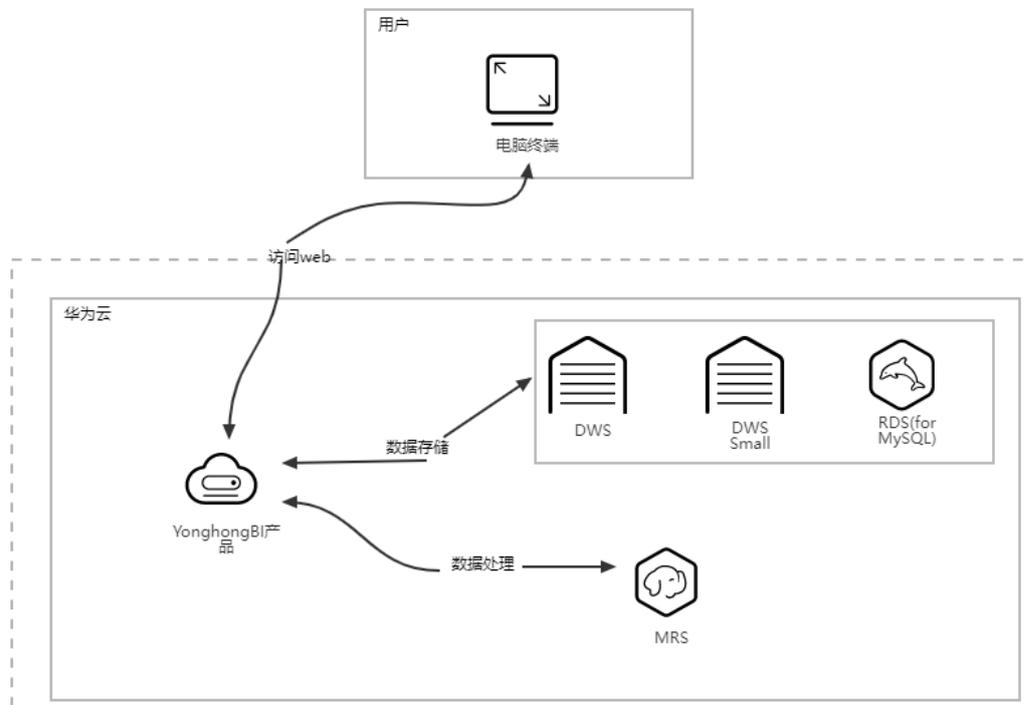
三个层次：基础业务层自主分析快速汇报、中间管理层汇总预警及时处理、最高决策层全局把控科学决策；

四个改善：改善数据质量、改善业务流程规范、改善决策科学性和准确性、改善企业的可持续竞争力。

图 1-1 业务架构



图 1-2 部署架构



架构描述:

1. 方案由华为云弹性云服务器ECS等计算，存储和网络资源为基础，部署永洪商业智能软件

2. 方案采用了华为云数据仓库DWS或云数据库RDS for MySQL作数据存储，采用MapReduce服务 MRS企业级大数据平台来作数据分析

## 方案优势

商业智能解决方案把大数据分析所需的产品功能全部融入一个平台进行统一管控。为各种规模的企业提供灵活易用的全业务链的大数据分析解决方案，让每一位用户轻松挖掘大数据价值，获取深度洞察力。

1. 全方位数据源接入：提供全方面的业务数据源接入能力，包括全部主流关系型数据库，国产数据库，各类大数据平台和多维数据库等。
2. 快速安装部署：降低系统安装门槛，实现快速部署、快速应用、快速迭代，操作简单，系统自动处理数据。
3. 一站式操作：提供可视化流程式数据治理能力、自助式分析模式，简单易操作的属性，让业务人员通过拖拉拽的方式完成数据分析的全流程，所有分析过程可视化，从而实现以业务为主导的数据体系。
4. 智能化分析：通过增强分析，可以大大缩短数据分析时间，帮助用户洞察数据，即使是普通业务人员也能通过简单操作挖掘数据价值

# 2 资源和成本规划

永洪商业智能解决方案基于华为云部署的场景，根据客户需求，选择对应的永洪软件产品和规格，以及对应所需华为云资源清单如下表所示：

表 2-1 资源和成本规划

| 商品名称                      | 规格                 | 购买数量   | 包年（元/年）        |
|---------------------------|--------------------|--|----------------|
| 小微企业体验版/中型企业体验版/小微企业体验永久版 | 【应用服务器】ECS         | 永洪主机配置 X86计算   通用计算增强型   c6.xlarge.2   4核   8GB   系统盘 超高IO 100GB   数据盘 高IO 100GB | 1<br>¥552.20   |
|                           | 【网络】弹性公网IP         | 全动态BGP   独享   按带宽计费   5Mbit/s  | 2<br>¥230.00   |
|                           | 【数据仓库】DWS（可选）      | 100 GB * 2（副本）* 3 节点; dwsk2.xlarge 标准数仓 鲲鹏 4 vCPUs   32 GB * 3 节点;               | 1<br>¥2,973.60 |
|                           | 【数据库】RDS MySQL（可选） | X86   独享型   4核16GB   2个节点; 40GB;   | 1<br>¥1,454.00 |

| 商品名称                      | 规格   | 购买数量 | 包年<br>(元/年) |
|---------------------------|--|------|-------------|
|                           | <p>【大数据】MRS (可选)</p> <p>Master 节点: MRS服务管理费用 *2</p> <p>Master 节点: 规格:X86计算   通用计算增强型   c3ne.xlarge.4   4核   16GB *2</p> <p>Master 节点: 系统盘:高IO   480GB *2</p> <p>Master 节点: 数据盘:高IO   200GB *2</p> <p>node_group_1: MRS服务管理费用 *3</p> <p>node_group_1: 规格:X86计算   通用计算增强型   c3ne.2xlarge.2   8核   16GB *3</p> <p>node_group_1: 系统盘:高IO   480GB *3</p> <p>node_group_1: 数据盘:高IO   100GB   1个 *3</p> | 1    | ¥62,558.50  |
| 小微企业基础版/中型企业基础版/小微企业永久基础版 | <p>【应用服务器】ECS</p> <p>永洪主机配置  X86计算   通用计算增强型   c6.xlarge.2   8核   16GB   系统盘超高IO 100GB   数据盘 高IO 100GB</p>   | 1    | ¥843.00     |
|                           | <p>【网络】弹性公网IP</p> <p>全动态BGP   独享   按带宽计费   5Mbit/s</p>   | 2    | ¥230.00     |
|                           | <p>【数据仓库】DWS (可选)</p> <p>100 GB * 2 (副本) * 3 节点; dwsk2.xlarge 标准数仓 鲲鹏 4 vCPUs   32 GB * 3 节点;</p>  | 1    | ¥2,973.60   |
|                           | <p>【数据库】RDS MySQL (可选)</p> <p>X86   独享型   4核16GB   2个节点; 40GB;</p>   | 1    | ¥1,454.00   |

| 商品名称  | 规格   | 购买数量 | 包年 (元/年)   |
|---|--|------|------------|
|   | <b>【大数据】MRS (可选)</b><br>Master 节点: MRS服务管理费用 *2<br>Master 节点: 规格:X86计算   通用计算增强型   c3ne.xlarge.4   4核   16GB *2<br>Master 节点: 系统盘:高IO   480GB *2<br>Master 节点: 数据盘:高IO   200GB *2<br>node_group_1: MRS服务管理费用 *3<br>node_group_1: 规格:X86计算   通用计算增强型   c3ne.2xlarge.2   8核   16GB *3<br>node_group_1: 系统盘:高IO   480GB *3<br>node_group_1: 数据盘:高IO   100GB   1个 *3 | 1    | ¥62,558.50 |
| 小微企业高级版/中型企业高级版/小微企业永久高级版/小微企业增强版/中型企业增强版/小微企业永久增强版/大型企业基础版 | <b>【应用服务器】ECS</b><br>永洪主机配置  X86计算   通用计算增强型   c6.xlarge.2   16核   32GB   系统盘 超高IO 100GB   数据盘 高IO 100GB   | 1    | ¥1,551.00  |
|   | <b>【网络】弹性公网IP</b><br>全动态BGP, 独享, 10M   | 2    | ¥1,030.00  |
|   | <b>【数据仓库】DWS (可选)</b><br>100 GB * 2 (副本) * 3 节点; dwsk2.xlarge 标准数仓 鲲鹏 4 vCPUs   32 GB * 3 节点;  | 1    | ¥2,973.60  |
|   | <b>【数据库】RDS MySQL (可选)</b><br>X86   独享型   4核16GB   2个节点; 40GB;   | 1    | ¥1,454.00  |

| 商品名称            | 规格   | 购买数量 | 包年 (元/年)   |
|-----------------|--|------|------------|
|                 | <p>【大数据】MRS (可选)</p> <p>Master 节点: MRS服务管理费用 *2</p> <p>Master 节点: 规格:X86计算   通用计算增强型   c3ne.xlarge.4   4核   16GB *2</p> <p>Master 节点: 系统盘:高IO   480GB *2</p> <p>Master 节点: 数据盘:高IO   200GB *2</p> <p>node_group_1: MRS服务管理费用 *3</p> <p>node_group_1: 规格:X86计算   通用计算增强型   c3ne.2xlarge.2   8核   16GB *3</p> <p>node_group_1: 系统盘:高IO   480GB *3</p> <p>node_group_1: 数据盘:高IO   100GB   1个 *3</p> | 1    | ¥62,558.50 |
| 大型企业高级版/大型企业永久版 | <p>【应用服务器】ECS</p> <p>永洪主机配置  X86计算   通用计算增强型   c6.xlarge.2   16核   32GB   系统盘 超高IO 100GB   数据盘 高IO 100GB</p>   | 1    | ¥1,551.00  |
|                 | <p>【网络】弹性公网IP</p> <p>全动态BGP, 独享, 10M</p>   | 2    | ¥1,030.00  |
|                 | <p>【数据仓库】DWS (可选)</p> <p>100 GB * 2 (副本) * 3 节点; dwsk2.xlarge 标准数仓 鲲鹏 4 vCPUs   32 GB * 3 节点;</p>  | 1    | ¥2,973.60  |
|                 | <p>【数据库】RDS MySQL (可选)</p> <p>X86   独享型   4核16GB   3个节点; 80GB;</p>   | 1    | ¥4,554.00  |

| 商品名称    | 规格   | 购买数量 | 包年 (元/年)   |
|---------|--|------|------------|
|         | <p>【大数据】MRS (可选)</p> <p>Master 节点: MRS服务管理费用 *2</p> <p>Master 节点: 规格:X86计算   通用计算增强型   c3ne.xlarge.4   4核   16GB *2</p> <p>Master 节点: 系统盘:高IO   480GB *2</p> <p>Master 节点: 数据盘:高IO   200GB *2</p> <p>node_group_1: MRS服务管理费用 *3</p> <p>node_group_1: 规格:X86计算   通用计算增强型   c3ne.2xlarge.2   8核   16GB *3</p> <p>node_group_1: 系统盘:高IO   480GB *3</p> <p>node_group_1: 数据盘:高IO   100GB   1个 *3</p> | 1    | ¥62,558.50 |
| 集团企业基础版 | <p>【应用服务器】ECS</p> <p>永洪主机配置  X86计算   通用计算增强型   c6.xlarge.2   16核   32GB   系统盘 超高IO 100GB   数据盘 高IO 100GB</p>   | 1    | ¥1,551.00  |
|         | <p>【网络】弹性公网IP</p> <p>全动态BGP, 独享, 10M</p>   | 2    | ¥1,030.00  |
|         | <p>【数据仓库】DWS (可选)</p> <p>100 GB * 2 (副本) * 4 节点; dwsk2.xlarge 标准数仓 鲲鹏 4 vCPUs   32 GB * 4 节点;</p>  | 1    | ¥3,964.80  |
|         | <p>【数据库】RDS MySQL (可选)</p> <p>X86   独享型   4核16GB   4个节点; 100GB;</p>  | 1    | ¥6,200.00  |

| 商品名称              | 规格   | 购买数量 | 包年<br>(元/年) |
|-------------------|--|------|-------------|
|                   | <p>【大数据】MRS (可选)</p> <p>Master 节点: MRS服务管理费用 *2</p> <p>Master 节点: 规格:X86计算   通用计算增强型   c3ne.xlarge.4   4核   16GB *2</p> <p>Master 节点: 系统盘:高IO   480GB *2</p> <p>Master 节点: 数据盘:高IO   200GB *2</p> <p>node_group_1: MRS服务管理费用 *3</p> <p>node_group_1: 规格:X86计算   通用计算增强型   c3ne.2xlarge.2   8核   16GB *3</p> <p>node_group_1: 系统盘:高IO   480GB *3</p> <p>node_group_1: 数据盘:高IO   100GB   1个 *3</p> | 1    | ¥62,558.50  |
| 集团企业高级版/集团企业永久高级版 | <p>【应用服务器】ECS</p> <p>永洪主机配置  X86计算   通用计算增强型   c6.xlarge.2   16核   32GB   系统盘 超高IO 100GB   数据盘 高IO 100GB</p>   | 1    | ¥1,551.00   |
|                   | <p>【网络】弹性公网IP</p> <p>全动态BGP, 独享, 10M</p>   | 2    | ¥1,030.00   |
|                   | <p>【数据仓库】DWS (可选)</p> <p>100 GB * 2 (副本) * 5 节点; dwsk2.xlarge 标准数仓 鲲鹏 4 vCPUs   32 GB * 5 节点;</p>  | 1    | ¥4,956.00   |
|                   | <p>【数据库】RDS MySQL (可选)</p> <p>X86   独享型   4核16GB   5个节点; 120GB;</p>  | 1    | ¥7,910.00   |

| 商品名称 | 规格  | 购买数量 | 包年<br>(元/年) |
|------|---|------|-------------|
|      | <p>【大数据】<br/>MRS (可选)</p> <p>Master 节点: MRS服务管理费用 *2</p> <p>Master 节点: 规格:X86计算   通用计算增强型   c3ne.xlarge.4   4核   16GB *2</p> <p>Master 节点: 系统盘:高IO   480GB *2</p> <p>Master 节点: 数据盘:高IO   200GB *2</p> <p>node_group_1: MRS服务管理费用 *3</p> <p>node_group_1: 规格:X86计算   通用计算增强型   c3ne.2xlarge.2   8核   16GB *3</p> <p>node_group_1: 系统盘:高IO   480GB *3</p> <p>node_group_1: 数据盘:高IO   100GB   1个 *3</p> | 1    | ¥62,558.50  |

# 3 实施步骤

## 产品安装

- 安装环境

表 3-1 安装环境

|      |   |
|------|---|
| 操作系统 | 32位和64位的Windows、Linux、Mac OS。<br>说明：Linux的操作系统包括CentOS、Red Hat、Ubuntu、中科方德、统信、万里红、中标麒麟、银河麒麟、华为欧拉。 |
| 浏览器  | 支持市面上的主流浏览器：chrome、火狐、IE等，推荐chrome。   |
| JDK  | JDK1.9及以上，推荐JDK11。  |

- 安装步骤

远程登录弹性云服务区，按照下面步骤安装到云服务器的数据盘里。

**步骤1** 使用非root权限的yhuser用户上传获取到的产品安装包到服务器目录“/home/yhuser/BI”。

**步骤2** 进入产品安装包所在目录“/home/yhuser/BI”，执行chmod +x "Yonghong Z-Suite V10.0.sh"为产品添加可执行权限。

图 3-1 添加可执行权限

```
[yhuser@MPP4 ~]$ cd /home/yhuser/BI/  
[yhuser@MPP4 BI]$ chmod +x "Yonghong Z-Suite V10.0.sh"
```

**步骤3** 执行 sh "Yonghong Z-Suite V10.0.sh" 脚本，开始安装产品。

图 3-2 开始安装

```
[yhuser@MPP4 BI]$ sh "Yonghong Z-Suite V10.0.sh"  
Starting Installer ...
```

**步骤4** 设置产品使用的语言，输入1。

图 3-3 设置语言

```
Please select a language:  
1: Chinese (Simplified)  
2: English [Enter]  
3: Japanese  
[1 - 3]  
1
```

步骤5 确认将产品安装到您的计算机上，输入o。

图 3-4 确认安装

```
安装助手会将 Yonghong Z-Suite 安装到您的计算机上。  
确定 [o, 回车键], 取消 [c]  
o
```

步骤6 将产品安装到新目录，输入2。

图 3-5 安装目录

```
发现一个已经现有的安装。您是否希望升级该安装?  
是, 升级现有的安装 [1, 回车键]  
不, 安装至另一目录 [2]  
2
```

步骤7 查看产品的许可协议，并接受协议，输入1。



### 📖 说明

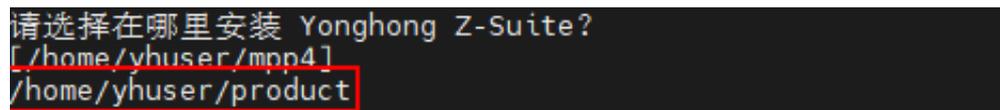
产品在license校验时会上报后台数据到永洪服务器，不会有其他任何形式的数据上报。

后台数据格式如下：`http://www.yonghongtech.com:8089/bi/validLicense?`

`license=xxx&ip=xxx&mac=xxx`，其中license即永洪提供给您的license号，ip是您的服务器地址，mac是您的服务器的物理地址。

- 步骤9** 设置产品的安装路径。安装路径默认为之前安装过的路径，用户可以自定义安装路径，此处设置为“`/home/yhuser/product`”。

图 3-8 设置安装路径

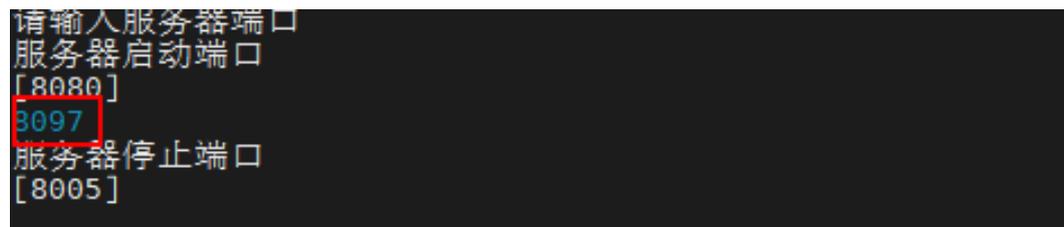


- 步骤10** 设置程序服务器启动端口号、停止端口号。

启动端口号默认为 8080，用户可以自定义端口号，此处设置为“8097”。

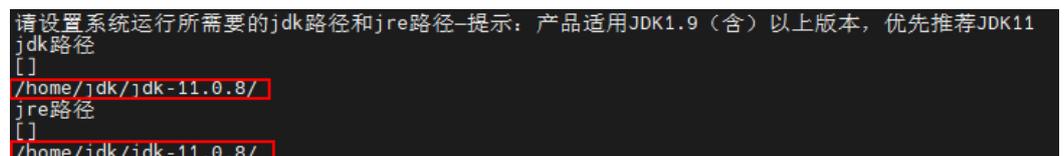
停止端口号默认为 8005，用户可以自定义端口号，此处采用默认，按下enter即可。

图 3-9 设置端口号



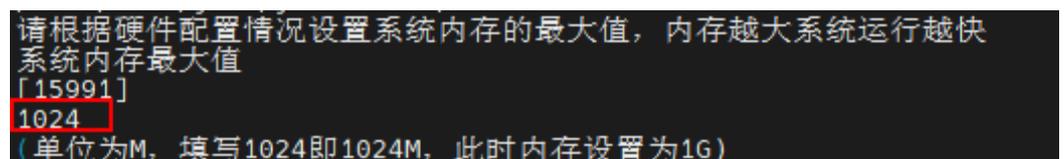
- 步骤11** 设置系统所需的Java环境的JDK、JRE路径，两者路径相同。产品适用JDK9（含）以上的版本，优先推荐JDK11

图 3-10 设置 JDK、JRE 路径



- 步骤12** 设置系统运行所需的最大内存，此处设置为“1024”。

图 3-11 设置内存



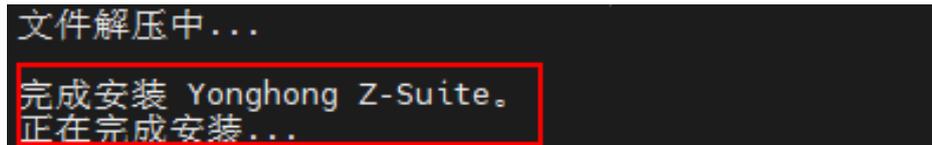
- 步骤13** 设置admin账号初始密码。密码需包括大小写字母和数字，且密码长度为8~32位。用户可根据实际情况设置希望的密码，此处设置为Admin1234。

图 3-12 设置初始密码



步骤14 等待产品安装完成。

图 3-13 等待安装完成



----结束

- 启动产品

进入产品安装包所在目录，如 “/home/yhuser/Yonghong Z-Suite/tomcat/bin/”  
“cd /home/yhuser/Yonghong Z-Suite/tomcat/bin/”

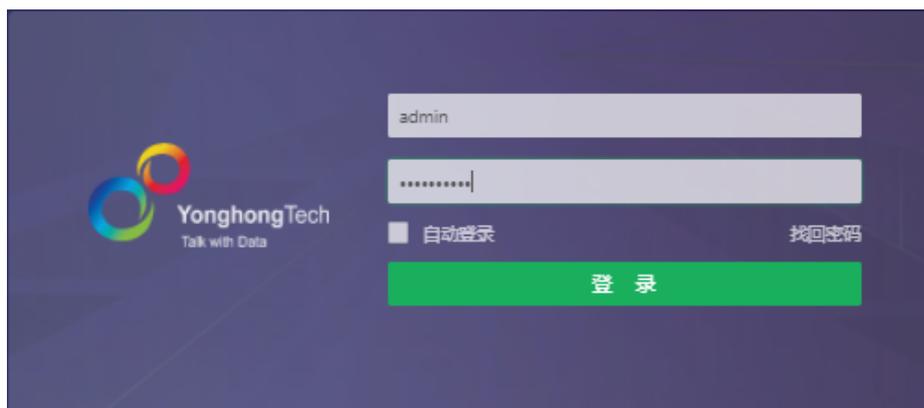
启动产品

```
sh startup.sh
```

- 登录产品

输入admin，安装时设置的密码，单击登录。

图 3-14 登录



## 集成 DWS

1. 新建一个DWS数据源，输入连接信息，保存数据源。
2. 测试连接，测试连接成功。

## 验证是否集成 DWS

1. 新建一个DWS数据源，输入连接信息，保存数据源，测试连接，测试连接成功。
2. 新建一个SQL数据集，选择创建好的DWS数据源，选择一个表，刷新数据，表内的前200条数据集被读取。保存数据集。
3. 新建一个报告，拖入一个表格组件，绑定步骤2里保存的数据集
4. 预览报告，显示表内所有数据。

## 释放资源

1. 远程登录弹性云服务器。
2. 执行命令：`sh /home/yhuser/product/uninstall.sh`，卸载永洪BI。卸载成功。
3. 释放弹性云服务器。

# 4 附录

## 4.1 背景信息

### 4.2 常见问题

## 4.1 背景信息

表 4-1 资源介绍

| 资源                  | 简介   |
|---------------------|--|
| Yonghong Z-Suite    | 一站式大数据分析平台，连续五年荣获敏捷BI领域第一名，中国BI大数据领域领军企业奖，10000+企业客户的共同选择。全面覆盖数据分析过程中的各个环节，包括数据采集、清洗、整合、存储、计算、建模、训练、展现、协作等，让用户可以在一个统一的平台上完成全流程数据分析任务，极大降低了实施、集成、培训的成本，帮助企业轻松构建数据应用。  |
| 弹性云服务器              | 弹性云服务器（Elastic Cloud Server, ECS）是由CPU、内存、操作系统、云硬盘组成的基础的计算组件。弹性云服务器创建成功后，您就可以像使用自己的本地PC或物理服务器一样，在云上使用弹性云服务器。<br>弹性云服务器的开通是自助完成的，您只需要指定CPU、内存、操作系统、规格、登录鉴权方式即可，同时也可以根据您的需求随时调整弹性云服务器的规格，为您打造可靠、安全、灵活、高效的计算环境。 |
| 数据仓库服务 GaussDB(DWS) | 数据仓库服务（Data Warehouse Service, 简称DWS）是完全托管的企业级云上数据仓库服务，具备免运维、在线扩展、高效的多源数据加载能力，兼容PostgreSQL生态。助力企业经济高效地对海量数据进行在线分析，实现数据快速变现。  |

## 4.2 常见问题

## 4.2.1 安装时遇到的问题

### 操作系统版本检查

- 执行以下命令可查看所有Linux的版本信息

```
lsb_release -a
```

正确输出信息如下：

```
[root@localhost bin]# lsb_release -a
LSB Version:      :core-4.1-amd64:core-4.1-noarch
Distributor ID:   CentOS
Description:      CentOS Linux release 7.5.1804 (Core)
Release:          7.5.1804
Codename:         Core
```

```
cat /etc/issue
```

正确输出信息如下：

```
[root@MPP7 ~]# cat /etc/issue
CentOS release 6.5 (Final)
Kernel \r on an \m
```

- 执行以下命令可查看系统的Linux版本信息

```
cat /etc/redhat-release
```

```
[root@localhost bin]# cat /etc/redhat-release
CentOS Linux release 7.5.1804 (Core)
```

### JDK 版本检查

- 执行以下命令可查看JDK版本信息

```
java -version
```

正确输出信息如下：

```
[root@mpp142 ~]# /home/tool/jdk/jdk-11.0.1/bin/java -version
openjdk version "11.0.1" 2018-10-16
OpenJDK Runtime Environment 18.9 (build 11.0.1+13)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.1+13, mixed mode)
[root@mpp142 ~]#
```

如无法输出上图所示信息或报错，请确认操作系统是否安装Java或者Java环境变量配置是否正确。

- 使用Open JDK的用户需要执行以下命令，检查开源字体是否安装。

```
fc-list
```

正确输出信息如下：

```
[root@localhost ~]# fc-list
/usr/share/fonts/ucs-miscfixed/6x12.bdf: Fixed:style=SemiCondensed
/usr/share/fonts/dejavu/DejaVuSerif-Bold.ttf: DejaVu Serif:style=Bold
/usr/share/fonts/paratype-pt-sans/PTS76F.ttf: PT Sans:style=Bold Italic
/usr/share/fonts/default/Type1/c059016l.pfb: Century Schoolbook L:style=Bold
/usr/share/fonts/ucs-miscfixed/6x13.bdf: Fixed:style=SemiCondensed
/usr/share/fonts/liberation/LiberationSerif-Italic.ttf: Liberation Serif:style=
/usr/share/fonts/open-sans/OpenSans-Semibold.ttf: Open Sans,Open Sans Semibold:
/usr/share/fonts/dejavu/DejaVuSerif-Italic.ttf: DejaVu Serif:style=Italic
/usr/share/fonts/open-sans/OpenSans-LightItalic.ttf: Open Sans,Open Sans Light:
/usr/share/fonts/default/Type1/c059033l.pfb: Century Schoolbook L:style=Italic
```

命令执行无输出或报错，说明系统未安装开源字体，请执行yum -install fontconfig命令安装字体。

### 📖 说明

Open JDK未安装开源字体导致图形界面无法调出，进而导致安装报错。

解决方案：如果服务器能使用yum命令，那么使用” yum install fontconfig “命令安装字体库即可；如果服务器无法使用yum命令，需要用户安装离线字体库，字体资源请到官网或其他渠道获取。

## 安装中遇到的问题

安装过程中，可能会出现一些问题导致异常、报错而导致安装失败。这里整理了一些问题和解决方案以供参考。

1. 安装报错 “java.lang.InternalError:  
java.lang.reflect.InvocationTargetException”

图 4-1 安装报错

```
[root@localhost ~]# sh Yonghong\ X+Viewer\ V9.4.1.sh
Starting Installer ...
java.lang.InternalError: java.lang.reflect.InvocationTargetException
    at java.desktop/sun.font.FontManagerFactory$.run(FontManagerFactory.java:86)
    at java.base/java.security.AccessController.doPrivileged(Native Method)
    at java.desktop/sun.font.FontManagerFactory.getInstance(FontManagerFactory.java:74)
    at java.desktop/sun.font.SunFontManager.getInstance(SunFontManager.java:247)
    at java.desktop/sun.font.FontDesignMetrics.getMetrics(FontDesignMetrics.java:265)
    at java.desktop/sun.swing.SwingUtilities2.getFontMetrics(SwingUtilities2.java:1231)
    at java.desktop/javafx.swing.JComponent.getFontMetrics(JComponent.java:1646)
    at java.desktop/javafx.swing.text.WrappedPlainView.updateMetrics(WrappedPlainView.java:443)
    at java.desktop/javafx.swing.text.WrappedPlainView.updateChildren(WrappedPlainView.java:422)
    at java.desktop/javafx.swing.text.WrappedPlainView.insertUpdate(WrappedPlainView.java:594)
    at java.desktop/javafx.swing.plaf.basic.BasicTextUI$RootView.insertUpdate(BasicTextUI.java:1709)
    at java.desktop/javafx.swing.plaf.basic.BasicTextUI$UpdateHandler.insertUpdate(BasicTextUI.java:1968)
    at java.desktop/javafx.swing.text.AbstractDocument.fireInsertUpdate(AbstractDocument.java:203)
    at java.desktop/javafx.swing.text.AbstractDocument.handleInsertString(AbstractDocument.java:757)
```

原因：服务器缺失字体库。

解决方案：如果服务器能使用yum命令，那么使用” yum install fontconfig “命令安装字体库即可；如果服务器无法使用yum命令，需要用户安装离线字体库，字体资源请到官网或其他渠道获取。

2. 安装过程中出现中文乱码

图 4-2 中文乱码

```
[root@localhost ~]# sh Yonghong\ X+Viewer\ V9.4.1.sh
Starting Installer ...
Please select a language:
1: Chinese (Simplified)
2: English [Enter]
3: Japanese
[1 - 3]
1
م«ë|àمèكُنēī لبا ۞م| Yonghong X+Viewer م«ë|àمèكُنēī ۞ |ه|«و»ن «ن ل ۞ه|ج|é
م»ل« [0, م ۞ل«و»ن «ن ل ۞ه|ج|é [c]
```

原因：可能是远程客户端编码非UTF-8。

解决方案：将编码设置为“UTF-8”，不同的客户端设置方法不同

### 3. 安装过程中提示系统语言不支持中文

图 4-3 不支持中文

```
[root@localhost ~]# sh Yonghong\ X+Viewer\ V9.4.1.sh
Starting Installer ...
Please select a language:
1: Chinese (Simplified)
2: English [Enter]
3: Japanese
[1 - 3]
1
Your current system charset does not support Chinese, please install the Chinese language pack and set the charset to zh_CN.
UTF-8 or use the English version product.
```

原因：当前系统语言环境非“zh\_CN.UTF-8”。

解决方案：

检查当前语言环境，命令：echo \$LANG；

语言环境改成“zh\_CN.UTF-8”，命令：export LANG=zh\_CN.UTF-8。

## 4.2.2 启动时遇到的问题

### 安装完成后，启动永洪 BI 失败

具体失败的现象如下：

- Windows环境，到tomcat/bin目录下，双击startup.bat启动永洪，tomcat弹窗闪退。
- Linux环境，到tomcat/bin目录下，执行sh startup.sh启动永洪，启动后查不到进程。

图 4-4 启动后查不到进程

```
[root@localhost bin]# pwd
/data/YH94X+Viewer/tomcat/bin
[root@localhost bin]# sh startup.sh
Using CATALINA_BASE:   /data/YH94X+Viewer/tomcat
Using CATALINA_HOME:   /data/YH94X+Viewer/tomcat
Using CATALINA_TMPDIR: /data/YH94X+Viewer/tomcat/temp
Using JRE_HOME:        /jdk11
Using CLASSPATH:       /data/YH94X+Viewer/tomcat/webapps/.../Yonghong/product/a
pi.jar:/data/YH94X+Viewer/tomcat/.../Yonghong/product/product.jar:/data/YH94X+Viewer/tomcat/.../Yonghong/product/thirds.jar:/d
ata/YH94X+Viewer/tomcat/.../Yonghong/lib/mongo-java-driver-3.11.0.jar:/data/YH94X+Viewer/tomcat/lib/servlet-api.jar:/data/YH9
4X+Viewer/tomcat/bin/bootstrap.jar:/data/YH94X+Viewer/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:   -Xmx521m -Xms521m -XX:MetaspaceSize=256m -XX:G1HeapRegionSize=32m -XX:InitiatingHeapOccupancyPercent
=35 -XX:G1ReservePercent=15 -XX:ConcGCThreads=10 -XX:ParallelGCThreads=10 -XX:+ParallelRefProcEnabled -XX:SoftRefLRUPolicyMS
PerMB=0 -Xlog:gc*:file=/data/YH94X+Viewer/tomcat/.../Yonghong/log/jvm.log:time:filecount=16 -Djava.awt.headless=true --add-e
xports java.base/jdk.internal.ref=ALL-UNNAMED --add-opens java.base/jdk.internal.ref=ALL-UNNAMED -XX:OnError="/jdk11/bin/jst
ack %p >/data/YH94X+Viewer/tomcat/logs/thread_stack_%p.log" -XX:ErrorFile=/data/YH94X+Viewer/tomcat/logs/hs_err_%p.log -XX:0
nOutOfMemoryError="/jdk11/bin/jmap -dump:format=b,file=/data/YH94X+Viewer/tomcat/logs/jmap_%p.log %p" -XX:+ExitOnOutOfMemory
Error -XX:-OmitStackTraceInFastThrow
Tomcat started.
[root@localhost bin]# ps -ef|grep tomcat
root      59498      1  0 16:21 pts/0    00:00:00 bash /data/YH94X+Viewer/tomcat/bin/reboot_main.sh /data/YH94X+Viewer/tomca
t/bin
root      59510  54977  0 16:21 pts/0    00:00:00 grep --color=auto tomcat
[root@localhost bin]#
```

如上截图，启动后查不到进程，这类现象都属于启动失败，进程没启动起来，这类问题的原因和解决方案可以参考如下三种：

1. JDK版本过低或JDK路径配置错误  
永洪所使用的JDK版本应高于JDK9，推荐JDK11，JDK版本过低会导致启动失败。检查tomcat/bin/setclassspath.sh（Windows环境检查setclassspath.bat）中JDK路径是否正确。

图 4-5 JDK 版本过低或 JDK 路径配置错误

```
[root@localhost bin]# head -4 setclassspath.sh
export JAVA_HOME=/usr/java/jdk-11.0.2
export JRE_HOME=/usr/java/jdk-11.0.2
#!/bin/sh

[root@localhost bin]# ll /usr/java/jdk-11.0.2/
总用量 28
drwxr-xr-x. 2 root root 4096 12月  1 20:19 bin
drwxr-xr-x. 4 root root  112 12月  1 20:19 conf
drwxr-xr-x. 3 root root  132 12月  1 20:19 include
drwxr-xr-x. 2 root root 4096 12月  1 20:19 jmods
drwxr-xr-x. 72 root root 4096 12月  1 20:19 legal
drwxr-xr-x. 8 root root 4096 12月 27 10:26 lib
-rw-r--r--. 1 root root  160 1月 18 2019 README.html
-rw-r--r--. 1 root root 1256 1月 18 2019 release
[root@localhost bin]#
```

### 说明

JDK11版本的JRE和JDK是一样的，其余版本不一定。

重新安装JDK，并配置环境变量，详情可参考[JAVA JDK11下载、安装与环境变量配置](#)。

2. 端口被占用  
这类原因一般可以在tomcat/logs/catalina.out启动日志中看到明显的报错提示。

图 4-6 报错提示 1

```
[root@localhost bin]# tail -50 ../logs/catalina.out
26-Jan-2022 16:38:14.665 信息 [main] org.apache.catalina.startup.VersionLoggerListener.log 命令行参数: -Dcatalina.bas
e=/data/YH94X+Viewer/tomcat
26-Jan-2022 16:38:14.665 信息 [main] org.apache.catalina.startup.VersionLoggerListener.log 命令行参数: -Dcatalina.hom
e=/data/YH94X+Viewer/tomcat
26-Jan-2022 16:38:14.665 信息 [main] org.apache.catalina.startup.VersionLoggerListener.log 命令行参数: -Djava.io.tmp
dir=/data/YH94X+Viewer/tomcat/temp
26-Jan-2022 16:38:14.666 信息 [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent 在java.library.path:[/usr/
java/packages/lib:/usr/lib64:/lib64:/lib:/usr/lib]上找不到基于APR的Apache Tomcat本机库。该库在边在生环环境中获得最佳性能
26-Jan-2022 16:38:16.448 信息 [main] org.apache.coyote.AbstractProtocol.init 初始化协议处理器 ["http-nio-8080"]
26-Jan-2022 16:38:16.542 严重 [main] org.apache.catalina.util.LifecycleBase.handleSubClassException 初始化组件[Connector[HTT
P/1.1-8080]]失败。
org.apache.catalina.LifecycleException: 协议处理程序初始化失败
    at org.apache.catalina.connector.Connector.initInternal(Connector.java:1049)
    at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:136)
    at org.apache.catalina.core.StandardService.initInternal(StandardService.java:556)
    at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:136)
    at org.apache.catalina.core.StandardServer.initInternal(StandardServer.java:1042)
    at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:136)
    at org.apache.catalina.startup.Catalina.load(Catalina.java:724)
    at org.apache.catalina.startup.Catalina.load(Catalina.java:746)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:566)
    at org.apache.catalina.startup.Bootstrap.load(Bootstrap.java:305)
    at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:475)
Caused by: java.net.BindException: 地址已在使用
    at java.base/sun.nio.ch.Net.bind0(Native Method)
    at java.base/sun.nio.ch.Net.bind(Net.java:461)
    at java.base/sun.nio.ch.Net.bind(Net.java:453)
    at java.base/sun.nio.ch.ServerSocketChannelImpl.bind(ServerSocketChannelImpl.java:227)
    at org.apache.tomcat.util.net.NioEndpoint.initServerSocket(NioEndpoint.java:271)
    at org.apache.tomcat.util.net.NioEndpoint.bind(NioEndpoint.java:226)
    at org.apache.tomcat.util.net.AbstractEndpoint.bindWithCleanup(AbstractEndpoint.java:1208)
    at org.apache.tomcat.util.net.AbstractEndpoint.init(AbstractEndpoint.java:1221)
    at org.apache.coyote.AbstractProtocol.init(AbstractProtocol.java:603)
    at org.apache.coyote.http11.AbstractHttp11Protocol.init(AbstractHttp11Protocol.java:80)
    at org.apache.catalina.connector.Connector.initInternal(Connector.java:1046)
    ... 13 more
```

解决方案：修改端口号。用文本编辑器打开Yonghong\tomcat\conf下有个server.xml文件，修改以下红色文字的端口号。

图 4-7 修改端口号

```
-->
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
            port="8080" protocol="HTTP/1.1"
-->
```

3. jvm内存过亿

这类原因一般可以在tomcat/logs/catalina.out启动日志中看到明显的报错提示。

图 4-8 报错提示 2

```
[root@localhost bin]# tail -f ../logs/catalina.out
java.base@11.0.2/java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(ScheduledThreadPoolExecutor.java:89
9)
java.base@11.0.2/java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1054)
java.base@11.0.2/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1114)
java.base@11.0.2/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)
java.base@11.0.2/java.lang.Thread.run(Thread.java:834)]
26-Jan-2022 16:42:52.806 信息 [Thread-28] org.apache.coyote.AbstractProtocol.stop 正在停止ProtocolHandler ["http-nio-8080"]
26-Jan-2022 16:42:52.814 信息 [Thread-28] org.apache.coyote.AbstractProtocol.destroy 正在摧毁协议处理器 ["http-nio-8080"]
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --a
dd-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi
.transport=ALL-UNNAMED
Error occurred during initialization of VM
Too small maximum heap
```

解决方案：调整分配给永洪jvm的内存

用文本编辑器打开Yonghong\tomcat\bin下有个catlina.sh文件（Windows环境修改catlina.bat），修改以下红色文字的内存大小，建议是服务器剩余可用内存的一半。

图 4-9 调整分配内存

```
if [ -z "$JSSE_OPTS" ]; then
  JSSE_OPTS="-Djdk.tls.ephemeralDHKeySize=2048"
fi
CATALINA_OPTS="-Xmx1024m -Xms1024m -XX:MetaspaceSize=256m -XX:G1HeapRegionSize=32m -XX:InitiatingHeapOccupancyPercent=35 -XX:G1ReservePercent=20 -XX:ParallelGCThreads=10 -XX:ParallelRefProcEnabled -XX:SoftRefLRUPolicyMSPerMB=0 -Xlog:gc*:file=$CATALINA_HOME/./Yonghong/log/jvm.log:time:filecount=16 -Djava.awt.headless=true -add-exports java.base/jdk.internal.ref=ALL-UNNAMED --add-opens java.base/jdk.internal.ref=ALL-UNNAMED -XX:OnErrors="$JAVA_HOME/bin/jstack %p >$CATALINA_HOME/logs/thread_stack_%p.log" -XX:ErrorFile=$CATALINA_HOME/logs/hs_err_%p.log -XX:OnOutOfMemoryError="$JAVA_HOME/bin/jmap -dump:format=b,file=$CATALINA_HOME/logs/jmap_%p.log %p" -XX:+ExitOnOutOfMemoryError -XX:-OmitStackTraceInFastThrow"
JAVA_OPTS="$JAVA_OPTS $JSSE_OPTS"
```

## 访问问题

永洪tomcat启动后，浏览器使用http://ip:port/bi/Viewer无法访问，可能的原因和解决方案有以下两种：

### 1. 网络不通

用浏览器所在的电脑，在cmd窗口使用“ping 服务器ip”命令，测试下网络是否通畅，如果网络不通，请联系自家网管。

如下图是网络通畅。

图 4-10 网络通畅

```
C:\Users\edz>ping 192.168.0.117

正在 Ping 192.168.0.117 具有 32 字节的数据:
来自 192.168.0.117 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.0.117 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.0.117 的回复: 字节=32 时间<1ms TTL=64
```

### 2. 端口不通

即使网络通了，如果端口没开放，也可能无法访问。

解决方案：关闭永洪服务器的防火墙或开放永洪tomcat访问端

# 5 修订记录

表 5-1 修订记录

| 发布日期       | 修订记录     |
|------------|----------|
| 2024-03-01 | 第一次正式发布。 |