|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| QQ截图20130904105420  **餐馆信息管理系统**  **系统安装部署手册**  设计人员： 2021112925-吴平凡  2021112923-袁旭阳  2021112937-李 品  指导教师 李 威   |  |  |  | | --- | --- | --- | | **文件状态：**  **[√ ] 草稿**  **[ ] 正式发布**  **[ ] 正在修改** | **文件标识** | **1.0** | | **当前版本** | **V1.0** | | **拟 稿 人** | 吴平凡 | | **拟稿日期** | 2024/6/14 | | **审 核 人** |  | | **审核日期** |  | |

编写说明

标题：系统安装部署手册

类别：文档

存放位置：项目文档\06、部署手册\系统安装部署手册.docx

编辑软件：Miscrosoft Word 2021 中文版

版本历史：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 日期 | 备注 |
| V1.0.1 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

完成情况分工：

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 工作量 | 完成工作内容 |
| 2021112923 | 吴平凡 | 70% | 完成了项目部署，并编写了报告 |
| 2021112925 | 袁旭阳 | 15% | 部署结果测试，报告的审核与修改 |
| 2021112937 | 李品 | 15% | 报告的审核与修改 |
|  |  |  |  |
|  |  |  |  |

目录

[编写说明 2](#_Toc169272227)

[1 编写目的 4](#_Toc169272228)

[1.1 读者对象 4](#_Toc169272229)

[1.2 参考资料 4](#_Toc169272230)

[2 安装部署要求 5](#_Toc169272231)

[2.1 服务器要求 5](#_Toc169272232)

[2.2 支撑软件安装要求 5](#_Toc169272233)

[2.3 应用软件部署要求 7](#_Toc169272234)

[3 部署方案 9](#_Toc169272235)

[3.1 环境部署 9](#_Toc169272236)

[3.2 系统部署 10](#_Toc169272237)

[3.3 项目容器化部署方案 13](#_Toc169272238)

# 编写目的

本手册旨在为实施人员提供一个详细且全面的指南，确保软件系统能够顺利安装、配置和部署到目标环境中。通过该手册，读者将能够理解系统部署的各个步骤和要求，从而保证系统的稳定运行和高效维护。本手册覆盖了从服务器要求到环境配置、应用部署以及部署后的验证和维护等各个方面的内容。

## 读者对象

本手册适用于以下人员：

1. **系统管理员**：负责服务器和网络环境的配置和维护，确保系统在硬件和操作系统层面的要求都能够满足。
2. **技术支持工程师**：负责为用户提供技术支持，处理系统运行中的问题和故障，并协助进行系统的安装和维护工作。
3. **实施人员**：直接参与系统的部署和实施，确保系统在客户环境中顺利上线并能稳定运行。
4. **相关技术人员**：包括开发人员、测试人员和运维人员，他们需要了解系统的部署细节，以便在需要时提供必要的支持和维护。

## 参考资料

为了确保部署过程的准确性和全面性，本手册参考了以下资料：

1. **《系统概要设计报告》**：提供了系统的功能介绍和操作指南，帮助实施人员理解系统的使用和功能。
2. **《系统详细设计报告》**：详细描述了系统的架构设计、模块划分和接口定义，帮助实施人员了解系统的技术细节和内部结构。
3. **《系统需求规格说明书》**：列出了系统的功能需求和性能要求，帮助实施人员明确系统需要实现的目标和性能指标。
4. **相关软件安装指南**：

**SpringBoot官方文档**：提供了SpringBoot的安装、配置和使用指南，帮助实施人员正确部署后端系统。

**QT官方文档**：提供了QT的安装、配置和使用指南，帮助实施人员正确部署客户端应用。

通过参考这些资料，实施人员可以获得必要的信息和指导，确保系统部署的各个环节都能够顺利进行。

# 安装部署要求

本节介绍了系统部署所需的硬件和软件要求，包括服务器要求、支撑软件安装要求和应用软件部署要求。确保在部署前满足这些要求，以保证系统的稳定运行和性能。

### 服务器要求

#### 硬件配置

* **CPU：**
  + **最少配置：4核**
  + **推荐配置：8核或以上**
* **内存：**
  + **最少配置：16GB**
  + **推荐配置：32GB或以上**
* **硬盘：**
  + **最少配置：500GB**
  + **推荐配置：1TB或以上**
* **网络：**
  + **千兆以太网**

#### 操作系统

* **Windows Server**：
  + Windows Server 2019或更高版本
* **Linux**：
  + Ubuntu 20.04 LTS或更高版本
  + CentOS 8或更高版本

### 2.2 支撑软件安装要求

为了确保系统的顺利运行，需预先安装以下支撑软件：

#### Java环境

* **JDK**：
  + 版本：JDK 17或以上版本
  + 安装方法：请参考[JDK官方安装文档](https://docs.oracle.com/en/java/javase/11/install/installation-jdk-macos.html)进行安装。

#### Web服务器

* **Apache Tomcat**：
  + 版本：Apache Tomcat 9或以上版本
  + 安装方法：请参考[Tomcat官方安装文档](https://tomcat.apache.org/tomcat-9.0-doc/setup.html)进行安装。

#### 数据库

* **MySQL**：
  + 版本：MySQL 8.0或以上版本
  + 安装方法：请参考[MySQL官方安装文档](https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/)进行安装。

#### 其他

* **Node.js**：
  + 版本：Node.js 14或以上版本
  + 安装方法：请参考Node.js官方安装文档进行安装。
* **Nginx**：
  + 版本：最新稳定版
  + 安装方法：请参考Nginx官方安装文档进行安装。
* **Git**：
  + 版本：最新稳定版
  + 安装方法：请参考[Git官方安装文档](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git" \t "_new)进行安装。
* **Docker**：
  + 版本：最新稳定版
  + 安装方法：请参考Docker官方安装文档进行安装。

### 2.3 应用软件部署要求

#### 客户端

* **QT版本**：
  + 版本：QT 5.15或以上
  + 安装方法：请参考QT官方安装文档进行安装和配置。

#### 后端系统

* **SpringBoot版本**：
  + 版本：SpringBoot 2.5或以上
  + 安装方法：请参考[SpringBoot官方文档](https://docs.spring.io/spring-boot/docs/current/reference/html/getting-started.html)进行项目构建和配置。

#### 配置要求

* **环境变量**：
  + 设置JAVA\_HOME，指向JDK安装路径
  + 设置CATALINA\_HOME，指向Tomcat安装路径
  + 设置NODE\_HOME，指向Node.js安装路径
  + 设置NGINX\_HOME，指向Nginx安装路径
* **配置文件**：
  + 确保所有配置文件（如application.properties或application.yml）都已正确配置，包括数据库连接、端口号、日志路径等。

#### 依赖安装

* **后端依赖**：
  + 使用Mave进行项目依赖管理
  + 执行mvn clean install或gradle build来下载和安装项目所需的所有依赖
* **前端依赖**：
  + 使用npm进行项目依赖管理
  + 执行npm install或yarn install来下载和安装项目所需的所有依赖

通过满足以上安装部署要求，可以确保系统在正式部署时能够顺利运行，并具备良好的性能和稳定性。

# 部署方案

### 3.1 环境部署

#### 操作系统安装

1. **安装和配置Windows Server / Ubuntu / CentOS**
   * **Windows Server**：
     + 插入Windows Server安装介质，启动计算机并按照屏幕上的指示进行安装。
     + 安装完成后，进行初步配置，如设置管理员密码、配置IP地址等。
   * **Ubuntu**：
     + 下载Ubuntu Server ISO文件，创建可启动的USB驱动器，启动计算机并按照屏幕上的指示进行安装。
     + 安装完成后，进行初步配置，如设置root密码、配置IP地址等。
   * **CentOS**：
     + 下载CentOS ISO文件，创建可启动的USB驱动器，启动计算机并按照屏幕上的指示进行安装。
     + 安装完成后，进行初步配置，如设置root密码、配置IP地址等。
2. **配置网络和防火墙**
   * 确保服务器具有静态IP地址。
   * 配置防火墙规则，允许必要的端口，如：
     + HTTP/HTTPS：80, 443
     + 数据库端口：3306（MySQL）
     + 应用程序端口：8080（Tomcat）
     + 其他必要的端口根据具体需求配置。

#### 支撑软件安装

1. **安装并配置JDK**
   * 下载JDK安装包并按照操作系统的说明进行安装。
   * 设置JAVA\_HOME环境变量，指向JDK安装路径。
   * 更新PATH环境变量，添加JAVA\_HOME/bin路径。
2. **安装并配置MySQL**
   * 下载MySQL安装包并按照操作系统的说明进行安装。
   * 运行mysql\_secure\_installation命令进行初步安全配置。
   * 创建用于应用程序的数据库和用户，并为用户赋予相应的权限。
3. **安装并配置Apache Tomcat**
   * 下载Apache Tomcat安装包并解压。
   * 设置CATALINA\_HOME环境变量，指向Tomcat安装路径。
   * 更新PATH环境变量，添加CATALINA\_HOME/bin路径。
   * 配置Tomcat服务器，如设置端口、配置用户等。
4. **安装并配置Git和Docker**
   * 下载并安装Git。
   * 设置GIT\_HOME环境变量，指向Git安装路径。
   * 更新PATH环境变量，添加GIT\_HOME/bin路径。
   * 下载并安装Docker。
   * 配置Docker以启动和管理容器。

### 3.2 系统部署

#### 数据库部署

配置数据库连接信息，按照下面的配置数据库连接信息

1. **URL**://mycgck.mysql.database.azure.com:3306/restaurantmanage?useSSL=false
2. **Username**: swjtuAdmin
3. **Password**: 77597759Abc
4. **修改后端配置文件**

在SpringBoot项目中，打开application.properties或application.yml文件，添加或修改以下配置：

spring.datasource.url=jdbc:mysql://mycgck.mysql.database.azure.com:3306/restaurantmanage?useSSL=false

spring.datasource.username=swjtuAdmin

spring.datasource.password=77597759Abc

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

#### 后端系统部署

1. **编译和打包SpringBoot应用**
   * 进入项目目录并使用Maven进行打包：

mvn clean package

1. **直接运行Jar包**

java -jar path\_to\_your\_application.jar

1. **配置应用的环境变量和配置文件**

根据部署环境，配置application.properties或application.yml文件，设置数据库连接、端口号、日志路径等。

#### 客户端部署

1. **安装QT客户端**
   * 根据操作系统下载并安装QT客户端。
2. **配置客户端连接参数**
   * 启动QT客户端，并在配置界面设置服务器地址、端口号和其他必要的连接参数。

通过以上步骤，可以完成系统的环境部署和系统部署，确保系统在目标环境中顺利运行。

### 3.3 项目容器化部署方案

容器化部署可以提高应用的可移植性和管理效率。以下是使用Docker进行容器化部署的详细步骤。

#### 3.3.1 安装Docker和Docker Compose

1. **安装Docker**
   * **Ubuntu**：

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update

sudo apt-get install -y docker-ce

* + **CentOS**：

sudo yum install -y yum-utils

sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

sudo yum install -y docker-ce

sudo systemctl start docker

sudo systemctl enable docker

* + **Windows**：
    - 下载并安装Docker Desktop Docker Desktop。

1. **安装Docker Compose**
   * **Linux**：

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

* + **Windows**：

Docker Desktop包含了Docker Compose，不需要额外安装。

#### 3.3.2 编写Dockerfile

1. **后端应用Dockerfile**

# 拉取jdk17作为基础镜像

FROM openjdk:17-jdk-buster

# 作者

MAINTAINER PingfanWu <friendfan154@gmail.com>

COPY restaurantManage.jar /app.jar

EXPOSE 8080

ENTRYPOINT ["java","-jar","app.jar"]**前端应用Dockerfile**（假设使用React）

#### 3.3.4 部署容器

1. **构建和启动容器**

在docker-compose.yml文件所在目录运行以下命令：

docker-compose up --build

1. **检查容器状态**

docker-compose ps

确保所有服务都在运行。

1. **访问应用**

<http://110.42.66.160:55441/>

#### 3.3.5 管理和维护容器

1. **停止容器**

docker-compose down

1. **重启容器**

docker-compose up

1. **查看日志**

docker-compose logs -f