RustChatGPT项目的部署计划，通过详细的步骤和回退计划，可以确保系统在生产环境中的顺利部署和稳定运行。

RustChatGPT部署计划

2024.7.3

摘要

彭铭琨、詹晓芹

小组成员

**修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | AMD | 修订者 | 说明 |
| V1.0 | 20240703 | A | 彭铭琨 | 新增部署计划 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

（A-添加，M-修改，D-删除）

目录

[修订历史记录 2](#_Toc12051)

[（A-添加，M-修改，D-删除） 2](#_Toc8357)

[1. 简介 4](#_Toc3693)

[项目名称：RustChatGPT 4](#_Toc15373)

[2. 部署目标 4](#_Toc21235)

[3. 部署环境 4](#_Toc24806)

[生产环境配置 5](#_Toc26958)

[网络配置 5](#_Toc31424)

[软件依赖 5](#_Toc16921)

[4. 部署前准备 5](#_Toc17906)

[准备工作 5](#_Toc1514)

[准备资源 5](#_Toc6124)

[1. 服务器配置：确保生产服务器已按要求配置。 5](#_Toc3803)

[5. 部署步骤 6](#_Toc31517)

[部署应用 6](#_Toc31575)

[安装Python依赖 6](#_Toc17179)

[6. 回退计划 6](#_Toc3111)

[7. 部署后的验证 6](#_Toc8938)

[验证步骤 6](#_Toc20505)

[8. 维护和支持 7](#_Toc24212)

[监控和报警 7](#_Toc29114)

[日常维护 7](#_Toc14964)

## 1. 简介

**项目名称**：RustChatGPT

**文档目的**：本部署计划旨在指导如何将RustChatGPT系统部署到生产环境，以确保系统的稳定运行和可用性。

**部署日期**：2024年7月05日

## 2. 部署目标

* 确保RustChatGPT系统在生产环境中的正常运行。
* 提供用户所需的全部功能，确保系统性能和安全性。
* 最小化部署过程中的风险和停机时间。

## 3. 部署环境

### 生产环境配置

* **服务器**：AWS EC2 实例
  + **操作系统**：Ubuntu 20.04
  + **CPU**：4 核
  + **内存**：16GB
  + **存储**：500GB SSD

### 网络配置

* **域名**：rustchatgpt.example.com
* **SSL证书**：Let's Encrypt

### 软件依赖

* **Web 服务器**：Nginx
* **应用服务器**：Gunicorn
* **数据库**：MySQL 8.0
* **缓存**：Redis
* **语言运行环境**：Python 3.8
* **框架**：FastAPI
* **前端**：Vue.js
* **包管理**：pip, npm

## 4. 部署前准备

### 准备工作

1. **备份数据**：确保所有现有数据已备份。
2. **测试环境验证**：在测试环境中验证部署过程。
3. **通知相关人员**：通知所有相关人员部署计划和时间表。

### 准备资源

1. **服务器配置**：确保生产服务器已按要求配置。
2. **域名和SSL证书**：确保域名解析正确，SSL证书已获取并配置。

## 5. 部署步骤

### 部署应用

**连接到生产服务器**

**更新系统和安装依赖**

**克隆代码库**

**安装Python依赖**

**前端构建**

**配置Gunicorn**

**配置Nginx**

**设置数据库**

**设置环境变量**

## 6. 回退计划

如果部署过程中出现严重问题，可以采取以下回退措施：

**停止当前部署**

**恢复备份**

**重新启动旧版本**

## 7. 部署后的验证

### 验证步骤

1. **访问主页**：确认主页正常加载。
2. **用户注册和登录**：测试用户注册和登录功能。
3. **聊天功能**：测试聊天功能，确保消息发送和接收正常。
4. **性能测试**：进行基本的性能测试，确认系统响应时间在预期范围内。
5. **安全性检查**：验证系统安全性，确保没有重大漏洞。

## 8. 维护和支持

### 监控和报警

* **监控工具**：使用Prometheus和Grafana进行系统监控。
* **报警设置**：设置关键指标的报警，如CPU使用率、内存使用率、API响应时间等。

### 日常维护

* **日志管理**：定期检查系统日志，解决潜在问题。
* **系统更新**：定期更新系统和依赖库，确保系统安全和稳定。
* **用户反馈**：收集用户反馈，持续改进系统。