Achieveit  
 软件部署文档

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： |  |
| 当前版本： | 1.0 |
| 作 者： | G03 |
| 完成日期： | 2020.04.17 |

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| V1.0 | G03 | 陈弈君  曹威杰 | 2020.04.17-  2020.04.17 |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 云服务器 3](#_Toc37992455)

[1.1 云服务器Spec 3](#_Toc37992456)

[1.2 端口配置 3](#_Toc37992457)

[2 数据库部署 3](#_Toc37992458)

[2.1 MySQL安装 3](#_Toc37992459)

[2.2 MySQL启动 4](#_Toc37992460)

[2.3 MySQL配置root远程访问 4](#_Toc37992461)

[2.4 MySQL数据表生成 4](#_Toc37992462)

[2.4.1 Navicat连接远程数据库 4](#_Toc37992463)

[2.4.2 Navicat导入数据结构表格 4](#_Toc37992464)

[3 后端部署 5](#_Toc37992465)

[3.1 后端部署流程概述 5](#_Toc37992466)

[3.2 Spring Boot中的Maven打包 5](#_Toc37992467)

[3.3 Intellij IDEA部署工具 5](#_Toc37992468)

[3.4 achieveit应用启动 6](#_Toc37992469)

[3.4.1 Intellij IDEA配置SSH 6](#_Toc37992470)

[3.4.2 achieveit启动脚本 6](#_Toc37992471)

# 云服务器

## 云服务器Spec

|  |  |
| --- | --- |
| 需求名称 | 详细要求 |
| OS | CentOS 7.5 64bit |
| CPU | 1 |
| 内存 | 2GB |
| 带宽 | 1MBps |

## 端口配置

|  |  |  |
| --- | --- | --- |
| 端口类型 | 协议 | 端口范围 |
| HTTP | TCP | 80 |
| HTTPS | TCP | 443 |
| SSH | TCP | 22 |
| MySQL | TCP | 3306 |
| Navicat | TCP | 8079 |

# 数据库部署

## MySQL安装

|  |
| --- |
| # wget http://repo.mysql.com//[mysql-ver-to-install].noarch.rpm  # rpm -ivh [mysql-ver-to-install].noarch.rpm  # yum install mysql-server  # yum install mysql-devel  # yum install mysql  # rpm -qa | grep -i mysql |

请在运行安装脚本前确保系统中wget和rpm运行正常，安装脚本如下

[mysql-ver-to-install]需要配置为需要的MySQL安装版本安装包前缀

## MySQL启动

|  |
| --- |
| # service mysqld start  # systemctl start mysqld |

以下为MySQL启动脚本

## MySQL配置root远程访问

进入数据库后执行以下命令更新root用户的访问authentication

|  |
| --- |
| mysql> use mysql;  mysql> show tables;  select user,authentication\_string from user;  update mysql.user set authentication\_string=password('your password') where user='root';  flush privileges;  exit |

## MySQL数据表生成

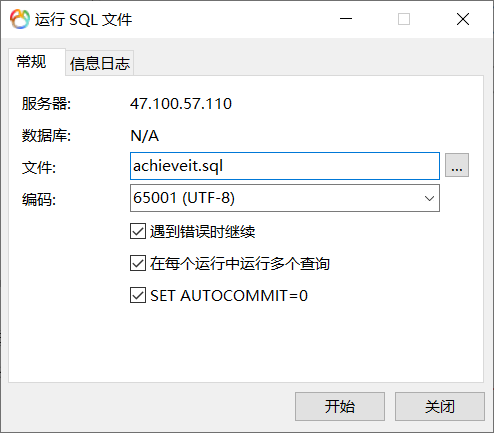
### Navicat连接远程数据库

由于开启了root用户的远程访问权限，可以用Navicat直接通过3306端口连接数据库而不用配置ssh，具体config如下图所示。



### Navicat导入数据结构表格

点击配置的连接中的[运行SQL文件]选项，选择附件中给出的achieveit.sql文件和[65001 (UTF-8)]编码方式，点击运行导入achieveit表及数据。具体见下图。



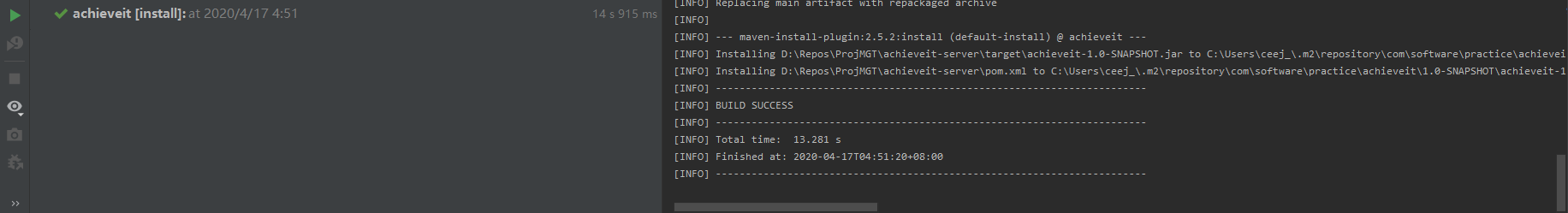
# 后端部署

## 后端部署流程概述

achieveit的后端基于Spring Boot开发，开发中使用Intellij IDEA中提供的工具设计的部署流程，流程分别为：Maven打包应用、sftp上传应用jar包、服务器启动程序。

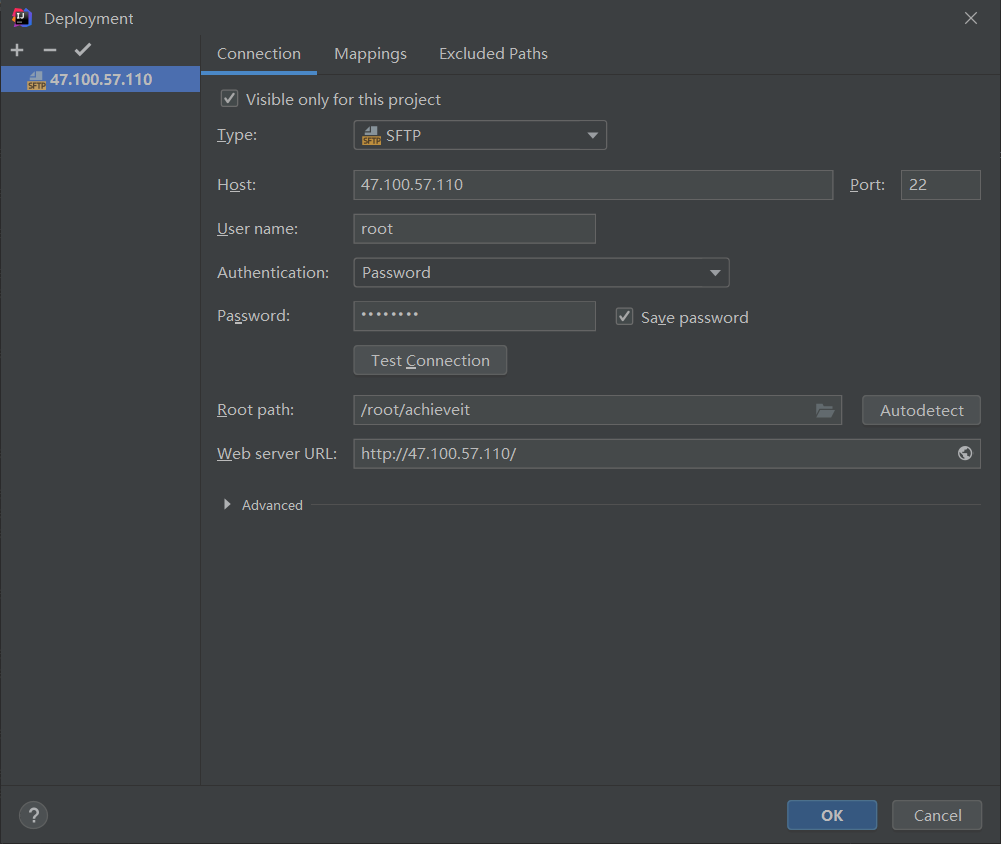
## Spring Boot中的Maven打包

Spring Boot应用程序可以使用Intellij中的Maven工具进行打包，打包仅需要点击Maven Tab中的install按钮就可以在target目录下生成[achieveit-ver-SNAPSHOT].jar文件。打包结果如下图所示。



## Intellij IDEA部署工具

Intellij IDEA提供了部署的一套解决方案来帮助开发者便捷地在开发环境中进行部署。这里我们使用其提供地SFTP类型来进行jar包传输，传输的地址映射为目标服务器的[/root/achieveit]为服务器期望的archiveit应用目录。具体配置如下

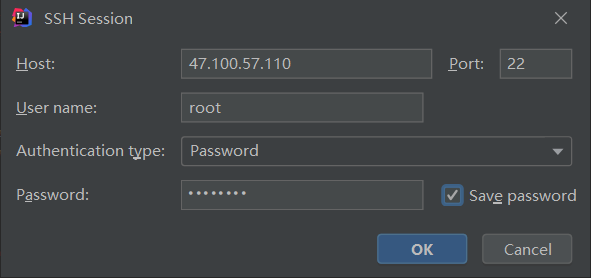


完成配置后，我们可以简单的对项目重target目录下的jar进行Deployement的上传。

## achieveit应用启动

### Intellij IDEA配置SSH

使用Intellij IDEA中自带的SSH工具可以提供在开发环境中对服务器的远程连接，SSH的配置如下：



### achieveit启动脚本

Spring Boot应用的启动需要依靠nohup程序来隐藏于后台运行。我们首先使用ps程序结合linux中的过滤器找出已经存在的achieveit进程，对其杀死后重新启动新部署的jar包，脚本具体如下所示。

|  |
| --- |
| #!/bin/bash  ps aux|grep achieveit|awk '{print $2}'|xargs kill -9  rm -R nohup.out  nohup java -jar achieveit-1.0-SNAPSHOT.jar |

在应用成功启动后可以得到nohup.out文件中的如下输出提示。

