

# Covid 19 疫情分析系统部署文档

## • 获取项目

确保网络通信良好，使用

```
1 | git clone https://github.com/BillChen2K/covid19
```

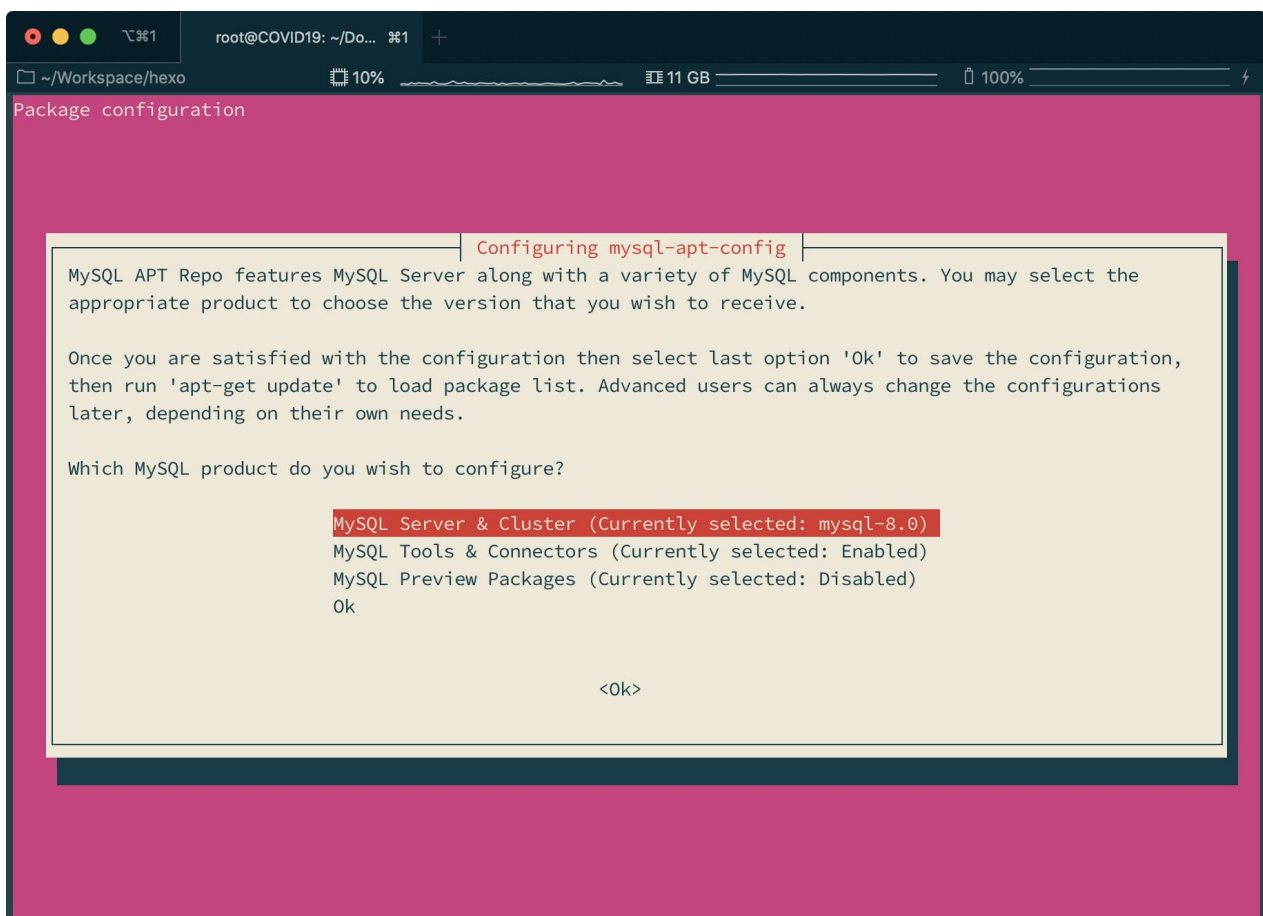
来讲项目保存到本地。前端和后端的源代码分别位于 `frontend` 目录和 `backend` 目录下。

## • 安装数据库

先初始化服务器，接下来使用

```
1 | wget https://dev.mysql.com/get/mysql-apt-config_0.8.15-1_all.deb
2 | dpkg -i mysql-apt-config_0.8.15-1_all.deb
```

安装 MySQL。在这里选择 8.0 版本：



```
1 | apt update
2 | apt install mysql-server
```

输入 `mysql --version` 检查版本，MySQL 8.0 安装成功。

```
1 | root@COVID19:~/Downloads# mysql --version
2 | mysql Ver 8.0.19 for Linux on x86_64 (MySQL Community Server - GPL)
```

在 MySQL 中授予远程访问权限：

```
1 | CREATE USER 'covid19'@'%' IDENTIFIED BY 'password';
2 | GRANT ALL PRIVILEGES ON *.* TO 'covid19'@'%' WITH GRANT OPTION;
3 | flush privileges;
```

## • 后端部署

首先配置跨站白名单。在后端源码 `config` 目录下有 `GlobalCorsConfig.java` 类：

```
1 | @Configuration
2 | public class GlobalCorsConfig implements WebMvcConfigurer {
3 |     @Override
4 |     public void addCorsMappings(CorsRegistry registry) {
5 |         registry.addMapping("/**")
6 |             .allowedOrigins("https://covid19.billc.io")
7 |             .allowedMethods("*")
8 |             .allowedHeaders("*")
9 |             .allowCredentials(true);
10 |
11 |     }
12 | }
```

将 `.allowedOrigins` 修改为前端部署的地址以接受带有凭据和 session 的请求。

Maven 生成 jar 包后在后端服务器上运行即可。

```
1 | nohup java -jar backend.jar &
```

## • 前端部署

首先安装依赖：

```
1 | yarn
```

或

```
1 | npm install
```

然后配置后端 API 的地址。在 `component/global/Config.vue` 下有 `apiurl` 导出项：

```
1 export {  
2   apiurl: https://covid19api.billc.io  
3 }
```

该地址修改为后端地址。地址不应以斜线结尾。接下来使用 `nuxt-ts` 生成静态的文件：

```
1 | nuxt-ts generate
```

生成的文件将位于 `frontend` 目录下的 `dist` 目录。使用 `rsync` 快速部署到云端：

```
1 | rsync -avzP --delete dist/ root@v2-hk.billc.io:/var/www/covid19/
```

接下来在浏览器中输入地址即可正常运行。