**检测修改文件并上传(摘至DeepSeek)**

|  |
| --- |
| Bash 在Linux中，你可以使用 `inotifywait` 监控文件或目录的变化，并结合 `rsync` 将变化的文件上传到指定的远程服务器。以下是一个完整的示例脚本： ### 脚本功能 1. 使用 `inotifywait` 监控指定目录的文件变化（如创建、修改、删除等）。 2. 当检测到变化时，使用 `rsync` 将变化的文件同步到远程服务器。  ### 脚本代码 |

|  |
| --- |
| Bash  #!/bin/bash  # 配置参数 WATCH\_DIR="/path/to/watch" # 要监控的目录 REMOTE\_USER="username" # 远程服务器的用户名 REMOTE\_HOST="remote\_host" # 远程服务器的地址 REMOTE\_DIR="/path/on/remote" # 远程服务器的目标目录  # 确保 inotifywait 和 rsync 已安装 if ! command -v inotifywait &> /dev/null || ! command -v rsync &> /dev/null; then  echo "inotifywait和rsync未安装，请先安装 inotify-tools和rsync,正在安装..." else  # 监控目录变化并同步到远程服务器  echo "开始监控目录: $WATCH\_DIR"  inotifywait -m -r -e create,modify,delete --format '%w%f' "$WATCH\_DIR" |   while read FILE  do  echo "检测到文件变化: $FILE"  rsync -avz --delete "$WATCH\_DIR/" "$REMOTE\_USER@$REMOTE\_HOST:$REMOTE\_DIR"  if [ $? -eq 0 ]; then  echo "文件已同步到远程服务器: $REMOTE\_HOST:$REMOTE\_DIR"  else  echo "文件同步失败！"  fi  done fi |

|  |
| --- |
| Bash ### 脚本说明 1. \*\*`inotifywait` 参数\*\*:  - `-m`：持续监控（不退出）。  - `-r`：递归监控目录及其子目录。  - `-e`：指定监控的事件（如 `create`、`modify`、`delete`）。  - `--format`：输出变化的文件路径。  2. \*\*`rsync` 参数\*\*:  - `-a`：归档模式，保留文件属性。  - `-v`：显示详细信息。  - `-z`：压缩传输。  - `--delete`：删除远程服务器上不存在于本地的文件。  3. \*\*循环监控\*\*:  - 使用 `while read FILE` 循环读取 `inotifywait` 的输出，并触发 `rsync` 同步。  ---  ### 使用方法 1. 将脚本保存为 `sync\_on\_change.sh`。 2. 赋予脚本执行权限：  ```bash  chmod +x sync\_on\_change.sh  ``` 3. 运行脚本：  ```bash  ./sync\_on\_change.sh  ```  ---  ### 注意事项 1. 确保本地和远程服务器之间配置了 SSH 免密登录，否则 `rsync` 需要手动输入密码。  - 配置方法：  ```bash  ssh-keygen -t rsa  ssh-copy-id $REMOTE\_USER@$REMOTE\_HOST  ``` 2. 如果监控的目录文件较多，`rsync` 可能会比较耗时，可以根据需要调整 `rsync` 的参数。  ---  ### 示例 假设： - 监控目录：`/home/user/files` - 远程服务器：`user@192.168.1.100:/backup/files`  脚本配置： ```bash WATCH\_DIR="/home/user/files" REMOTE\_USER="user" REMOTE\_HOST="192.168.1.100" REMOTE\_DIR="/backup/files" ```  运行脚本后，任何在 `/home/user/files` 中的文件变化都会自动同步到远程服务器的 `/backup/files` 目录。 |