# 109slurm集群服务器搭建白皮书



日期：2020.11.30

**前言**

这是一篇slurm服务器集群的安装教程。本文的目标是成为一篇快速安装教程。从未接触过slurm的新手可以在学习更复杂更深入的知识以前通过本文获得一些简明扼要的关于slurm的可实际操作的知识。

本文描述了一种干净利落但实用有效的slurm安装的方法。虽然我们不会按照索引把从A到Z开头的所有步骤都介绍一遍，但是我们也不会回避slurm安装过程中一些难理解的、凌乱的、非标准、不常用但是却可用或很有用的问题。包括自启动软链接添加，munge认证软件的安装等等。我们的讨论将围绕安装过程展开，而不是为了让读者对slurm这个软件有什么领悟，因此我们并没有按照传统的slurm教程的思路来撰写本文。本文没有深入的教学方法，没有讲解slurm集群计算实现方式，也没有论述slurm软件的优点。这并不是说这些东西是不重要的，而是说它们与某些人正在寻找的“快速教程（我是说本文）”无关。

能有多快呢？我不知道一个人是否能在3小时内完成一整个服务器计算集群的搭建，虽然我听有人说了解Linux的基本内容应该一个周末就够了。Slurm使用的标准——所有精准和复杂的定义都包括进去——只有几十页长。这可能是因为真正对slurm的大彻大悟（当它到来时），只需一个下午，尽管在那之前不知要花费多少个下午。这就是我的简单介绍。

感谢刘雄超师兄将slurm这个软件介绍给我们，以及各位同学可靠和优雅的实现了slurm集群环境搭建过程（本书讲解的过程即是此安装过程）

==================

目录

[一、 根据各节点IP修改hosts文件 6](#_Toc57629227)

[1. 安装net-tools 6](#_Toc57629228)

[2. 查看所有节点IP，并修改所有节点的hosts文件 6](#_Toc57629229)

[二、 安装SSH 6](#_Toc57629230)

[1. 安装ssh 6](#_Toc57629231)

[2. 查看ssh运行状态 6](#_Toc57629232)

[三、 安装NTP实现时间同步 6](#_Toc57629233)

[1. 方法1：直接修改时区 6](#_Toc57629234)

[2. 方法2：使用ntpdate工具 7](#_Toc57629235)

[四、 安装munge 7](#_Toc57629236)

[1. 源码编译安装（全部节点） 7](#_Toc57629237)

[2. 创建密钥（主节点） 7](#_Toc57629238)

[3. 新建用户并修改文件所属用户（全部节点） 7](#_Toc57629239)

[4. 拷贝密钥（主节点） 8](#_Toc57629240)

[5. 启动守护进程（全部节点） 8](#_Toc57629241)

[五、 安装slurm 8](#_Toc57629242)

[1. 源码编译安装（全部节点） 8](#_Toc57629243)

[2. 新建用户并修改文件所属用户（全部节点） 9](#_Toc57629244)

[3. 配置主节点 9](#_Toc57629245)

[4. 分发配置文件 12](#_Toc57629246)

[5. 启动集群 12](#_Toc57629247)

[六、 配置环境变量 12](#_Toc57629248)

[1. 添加路径到环境变量中 12](#_Toc57629249)

[2. 设置开机自己加载一遍profile文件 13](#_Toc57629250)

[七、 设置集群内部共享文件夹 13](#_Toc57629251)

[1. 配置主机服务器：（这里采用了管理节点，可以自由选择） 13](#_Toc57629252)

[2. 配置客户端计算机：(除了NFS服务器的所有子节点) 14](#_Toc57629253)

[八、 为slurm控制节点搭建邮件服务器 15](#_Toc57629254)

[1. 方法1：使用postfix 15](#_Toc57629255)

[2. 方法2：使用sendmail 15](#_Toc57629256)

[九、 在控制节点创建多用户，进行用户管理 16](#_Toc57629257)

[1. 添加用户 16](#_Toc57629258)

[2. 修改用户SSH登录时的默认位置 16](#_Toc57629259)

[十、 配制开机自启动服务器脚本 16](#_Toc57629260)

[1. 创建需要执行的脚本 17](#_Toc57629261)

[2. 将该脚本添加到rc.local中 17](#_Toc57629262)

[3. 配置开机自启动链接 18](#_Toc57629263)

[4. 启动服务并检查服务状态 19](#_Toc57629264)

[十一、 安装计算环境 19](#_Toc57629265)

[1. Java运行环境的安装 19](#_Toc57629266)

[2. Matlab安装 20](#_Toc57629267)

注意：除了个别需要切换用户的命令，大部分指令都需要在root权限下执行。

|  |
| --- |
| sudo su #su代表切换root用户，且保留在当前路径 |

## 根据各节点IP修改hosts文件

此处以计算节点node1的hosts为例，其中manage代表控制节点，其他node为计算节点。只需在本机的两行配置后加入集群内部所有主机列表即可，可实现根据主机名字识别IP功能。

### 安装net-tools

|  |
| --- |
| apt-get install net-tools -y |

### 查看所有节点IP，并修改所有节点的hosts文件

|  |
| --- |
| ifconfig |
| nano /etc/hosts |

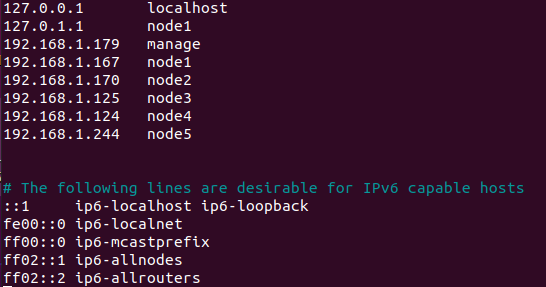
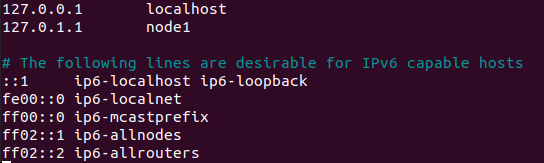


图1 (a)hosts文件修改前 (b)hosts文件修改后

## 安装SSH

使用SSH提供用户登录服务器的隧道，且后续安装过程中下达munge和slurm配置文件同需要SSH。

### 安装ssh

|  |
| --- |
| apt install ssh |

### 查看ssh运行状态

|  |
| --- |
| systemctl status ssh |

## 安装NTP实现时间同步

如果各节点时间不一致，会导致slurm各节点之间无法通信。

### 方法1：直接修改时区

直接修改时区

|  |
| --- |
| timedatectl set-timezone "Asia/Shanghai" |

### 方法2：使用ntpdate工具

1. 安装ntpdate工具

|  |
| --- |
| apt install ntpdate |

1. 将系统时间和网络时间同步

|  |
| --- |
| ntpdate cn.pool.ntp.org |

1. 将时间写入硬件

|  |
| --- |
| hwclock -systohc |

## 安装munge

munge提供集群间的认证通信机制，这个需要在所有节点安装并且启动。

参考链接：<https://www.cnblogs.com/haibaraai0913/p/11016885.html>

### 源码编译安装（全部节点）

|  |
| --- |
| #切换root  sudo su  #准备环境  apt install gcc  apt install openssl  apt install libssl-dev apt install make  #下载安装包  wget https://github.com/dun/munge/archive/munge-0.5.13.tar.gz  #解压  tar -zxf munge-0.5.13.tar.gz  #切换路径  cd munge-munge-0.5.13  #编译安装  ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var && make && make install |

### 创建密钥（主节点）

|  |
| --- |
| #等待随机数据 (推荐给偏执狂):  dd if=/dev/random bs=1 count=1024 >/etc/munge/munge.key  #获取为随机数据 (推荐给没有耐心的人):  dd if=/dev/urandom bs=1 count=1024 >/etc/munge/munge.key  #输入密码的散列 (不推荐):  echo -n "foo" | sha512sum | cut -d' ' -f1 >/etc/munge/munge.key  #直接输入密码 (真心不推荐):  echo "foo" >/etc/munge/munge.key |

### 新建用户并修改文件所属用户（全部节点）

|  |
| --- |
| #新建用户及其主目录和登录shell  useradd munge -m -s /bin/bash  #给用户赋密码  passwd munge  #修改目录属主  chown -R munge.munge /var/{lib,log,run}/munge  chown -R munge.munge /etc/munge  #修改目录模式  chmod 711 /var/lib/munge  chmod 700 /var/log/munge  chmod 755 /var/run/munge  chmod 700 /etc/munge  chmod 400 /etc/munge/munge.key |

### 拷贝密钥（主节点）

|  |
| --- |
| #拷贝主节点密钥到其余节点  scp /etc/munge/munge.key munge@ip:/etc/munge |

### 启动守护进程（全部节点）

|  |
| --- |
| #切换用户（root用户也可直接启动，不需要切换用户）  su munge  #启动munge  /etc/init.d/munge start #此处还有restart、stop等命令，可自行到相应目录查看 |

注意：关于munge的一些问题

|  |
| --- |
| 关于munge启动方式：   1. etc/init.d/munge start启动时会在/var/run/munge目录下创建munge.socket.2、munge.socket.2.lock、munged.pid，在/etc/init.d/munge stop时，这些文件会自行删除。 2. 如果利用munged启动munge服务，同样会生成以上三个文件，但是利用/etc/init.d/munge stop无法结束munge服务。如果之后还输入munged命令会出现以下错误【munged: Error: Found pid xxxx bound to socket "/var/run/munge/munge.socket.2".】这个问题重启可以解决（其实就是重启结束munge服务）所以建议使用以下方式开启munge服务/etc/init.d/munge start 3. [root@vm5 munge-0.5.10]# munge -n   munge: Error: Unable to access "/var/run/munge/munge.socket.2": No such file or directory  若遇到以上问题，则先 /etc/init.d/munge stop，然后/etc/init.d/munge start重启就ok了 |

## 安装slurm

参考链接：<https://www.cnblogs.com/haibaraai0913/p/11045295.html>

### 源码编译安装（全部节点）

|  |
| --- |
| #切换root  sudo su  #下载安装包  wget https://download.schedmd.com/slurm/slurm-19.05.0.tar.bz2  #解压  tar -xaf slurm\*tar.bz2  #切换路径  cd slurm-19.05.0  #编译安装  ./configure --enable-debug --prefix=/opt/slurm --sysconfdir=/opt/slurm/etc  make && make install |

注意：在编译过程中可能会出现的错误：/usr/bin/env："python"：没有那个文件或目录，解决方法：

|  |
| --- |
| #添加软链  ln -s /usr/bin/python3 /usr/bin/python |

### 新建用户并修改文件所属用户（全部节点）

|  |
| --- |
| #新建用户及其主目录和登录shell  useradd slurm -m -s /bin/bash  #给用户赋密码(此处需要输入两次slurm用户的密码)  passwd slurm #新建所需文件夹 mkdir /opt/slurm/log mkdir /opt/slurm/spool mkdir /opt/slurm/spool/slurm mkdir /opt/slurm/run #修改目录属主 chown -R slurm:slurm /opt/slurm |

### 配置主节点

|  |
| --- |
| #从源码包拷贝配置文件夹  cp -r /opt/package/slurm-19.05.0/etc/ /opt/slurm/etc/  #修改目录属主  chown -R slurm:slurm /opt/slurm/etc  #拷贝配置文件实例  cp /opt/slurm/etc/slurm.conf.example /opt/slurm/etc/slurm.conf  #打开配置文件进行编辑  nano /opt/slurm/etc/slurm.conf |

配置文件参数等详情请参考链接：

<https://slurm.schedmd.com/configurator.easy.html>

关于如何让控制节点参与计算，可参考链接：

<http://www.cocoachina.com/cms/wap.php?action=article&id=481922>

此处使用的配置文件：

|  |
| --- |
| # Example slurm.conf file. Please run configurator.html  # (in doc/html) to build a configuration file customized  # for your environment.  #  #  # slurm.conf file generated by configurator.html.  #  # See the slurm.conf man page for more information.  #  ClusterName=linux #集群名称  ControlMachine=control #控制节点名称  ControlAddr=192.168.0.104 #控制节点IP  #BackupController=  #BackupAddr=  #  SlurmUser=slurm  #SlurmdUser=root  SlurmctldPort=6817  SlurmdPort=6818  AuthType=auth/munge  #JobCredentialPrivateKey=  #JobCredentialPublicCertificate=  StateSaveLocation=/opt/slurm/spool/slurm/ctld  SlurmdSpoolDir=/opt/slurm/spool/slurm/d  SwitchType=switch/none  MpiDefault=none  SlurmctldPidFile=/opt/slurm/run/slurmctld.pid  SlurmdPidFile=/opt/slurm/run/slurmd.pid  ProctrackType=proctrack/pgid  #PluginDir=  #FirstJobId=  ReturnToService=0  #MaxJobCount=  #PlugStackConfig=  #PropagatePrioProcess=  #PropagateResourceLimits=  #PropagateResourceLimitsExcept=  #Prolog=  #Epilog=  #SrunProlog=  #SrunEpilog=  #TaskProlog=  #TaskEpilog=  #TaskPlugin=  #TrackWCKey=no  #TreeWidth=50  #TmpFS=  #UsePAM=  #  # TIMERS  SlurmctldTimeout=300  SlurmdTimeout=300  InactiveLimit=0  MinJobAge=300  KillWait=30  Waittime=0  #  # SCHEDULING  SchedulerType=sched/backfill  #SchedulerAuth=  #SelectType=select/linear  SelectType=select/cons\_tres #允许分配资源形式 Allocate individual processors, memory, GPUs, and other trackable resources  SelectTypeParameters=CR\_Core #以CPU作为分配资源(default) Cores as a consumable resource.  FastSchedule=1  #PriorityType=priority/multifactor  #PriorityDecayHalfLife=14-0  #PriorityUsageResetPeriod=14-0  #PriorityWeightFairshare=100000  #PriorityWeightAge=1000  #PriorityWeightPartition=10000  #PriorityWeightJobSize=1000  #PriorityMaxAge=1-0  #  # LOGGING  SlurmctldDebug=3  SlurmctldLogFile=/opt/slurm/log/slurmctld.log  SlurmdDebug=3  SlurmdLogFile=/opt/slurm/log/slurmd.log  JobCompType=jobcomp/none  #JobCompLoc=  #  # ACCOUNTING  #JobAcctGatherType=jobacct\_gather/linux  #JobAcctGatherFrequency=30  #  #AccountingStorageType=accounting\_storage/slurmdbd  #AccountingStorageHost=  #AccountingStorageLoc=  #AccountingStoragePass=  #AccountingStorageUser=  #  # COMPUTE NODES  #配置信息填写 在这里可设置节点资源（Procs为核心数）Weight为权重，权重序号越小执行优先级越高  NodeName=control Procs=6 State=IDLE Weight=2  NodeName=node1 Procs=6 State=IDLE Weight=1  #NodeName=node2 Procs=6 State=IDLE Weight=1  #添加节点，直接跟在Nodes就可以，Default代表参不参与计算，MaxTime是最大运行时间  PartitionName=control Nodes=control Default=Yes MaxTime=INFINITE State=UP  PartitionName=compute Nodes=node1,control Default=Yes MaxTime=INFINITE State=UP |

### 分发配置文件

注意：scp 默认不允许给另外一个root账号传输文件，传输对象必须是普通用户

|  |
| --- |
| #拷贝主节点配置节点到子节点  scp -r /opt/slurm/etc/ slurm@192.168.231.129:/opt/slurm/  scp -r /opt/slurm/etc/ slurm@192.168.231.130:/opt/slurm/ |

### 启动集群

|  |
| --- |
| #主节点root用户执行  /opt/slurm/sbin/slurmctld -c  /opt/slurm/sbin/slurmd -c  #子节点root用户执行  /opt/slurm/sbin/slurmd -c |

## 配置环境变量

配置环境变量可以将slurm的bin目录添加到环境目录中，即可以直接输入sinfo，squeue, scancel等常用命令，而不需要每次一都/opt/slurm/bin/sinfo来使用。

### 添加路径到环境变量中

参考链接：<https://blog.csdn.net/weixin_36296538/article/details/83044639?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.edu_weight>

|  |
| --- |
| #添加路径  nano /etc/profile  #在最后一行添加  export PATH=/opt/slurm/bin:$PATH |

### 设置开机自己加载一遍profile文件

参考链接：<https://blog.csdn.net/liu123342/article/details/109484621>

|  |
| --- |
| #在 ~/.bashrc 里面加一句 source /etc/profile ,然后执行一次 source ~/.bashrc 使该文件生效即可。  #编辑basharc文件  nano ~/.bashrc  #在该文件最后一行加一句 source /etc/profile  #执行一次使该文件生效  source ~/.bashrc |

## 设置集群内部共享文件夹

由于slurm会出现调用哪一个计算资源，就将输出文件放在哪一台节点上的问题，不便于管理，所以需要设置一个所有节点共享的文件夹。该文件夹后续可以建立不同用户的文件夹，通过设置权限可以对普通用户实现较好的隔离。

参考链接：<https://www.linuxidc.com/Linux/2018-11/155331.htm>

注意：每次关机后，都需要在计算节点上重新挂载一次共享文件夹。

### 配置主机服务器：（这里采用了管理节点，可以自由选择）

1. 安装NFS服务器

|  |
| --- |
| #更新Internet的存储库索引：  apt-get update  #在系统上安装NFS内核服务器：  apt install nfs-kernel-server -y |

1. 创建导出目录

我们要与客户端系统共享的目录称为导出目录。 你可以根据自己的选择来命名; 在这里，我们在控制节点的home目录中创建一个名为“share”的导出目录。

|  |
| --- |
| #创建需要共享的文件夹  mkdir -p /home/share  #通过以下命令删除文件夹的限制权限，保证所有用户都能访问  chown nobody:nogroup /home/share  chmod 777 /home/share |

1. 通过NFS导出文件为客户端分配服务器访问权限

创建导出文件夹后，我们需要为客户端提供访问主机服务器计算机的权限。 此权限是通过位于系统的/etc/exports文件定义的。

|  |
| --- |
| #编辑文件修改权限  nano /etc/exports   * #单个客户端通过在文件中添加以下行：   /home/share clientIP(rw,sync,no\_subtree\_check)   * #通过在文件中添加以下行来指定多个客户端：   /home/share client1IP(rw,sync,no\_subtree\_check)  /home/share client2IP(rw,sync,no\_subtree\_check)   * #多个客户端，通过指定客户端所属的整个子网：   /home/share 192.168.1.0/24(rw,sync,no\_subtree\_check) |

1. 导出共享目录

|  |
| --- |
| #在主机系统中完成上述所有配置后，现在可以通过以下命令将共享目录导出：  exportfs -a  #最后，为了使所有配置生效，请按如下方式重新启动NFS服务器：  systemctl restart nfs-kernel-server |

1. 为客户端打开防火墙

|  |
| --- |
| #配置防火墙以通过NFS授予客户端访问权限：  #ufw allow from [clientIP or clientSubnetIP] to any port nfs  #我们的示例中，我们通过以下命令访问客户端计算机的整个子网：  ufw allow from 192.168.1.0/24 to any port nfs  #检查Ubuntu防火墙的状态  ufw status |

### 配置客户端计算机：(除了NFS服务器的所有子节点)

1. 安装NFS Common

|  |
| --- |
| #更新我们系统的存储库索引和Internet的索引：  apt-get update  #在系统上安装NFS Common客户端：  apt-get install nfs-common |

1. 为NFS主机的共享文件夹创建安装点

客户端系统需要一个目录，可以访问导出文件夹中主机服务器共享的所有内容。您可以在系统的任何位置创建此文件夹。此处为了便于管理，我们选择了和服务器相同的目录。

|  |
| --- |
| #在客户端机器的home目录中创建一个share文件夹：  mkdir -p /home/share |

1. 在客户端上挂载共享目录

|  |
| --- |
| #使用以下命令将共享文件夹从主机安装到客户端上的装入文件夹：  #mount serverIP:/shareFolder\_server /mnt/mountfolder\_client，此处应该是：  #其中IP地址是NFS服务器地址  mount 192.168.182.172: /home/share /home/share |

1. 测试连接

请在NFS主机服务器的导出文件夹中创建或保存文件。现在，打开客户端计算机上的/home/share文件夹; 应该能够在此文件夹中查看共享和访问的同一文件。

## 为slurm控制节点搭建邮件服务器

由于有的程序运行时间较长，所以需要搭建邮件服务器，配合slurm的mail命令，可以在代码运行结束后以邮件的方式通知用户。

### 方法1：使用postfix

|  |
| --- |
| #在线安装postfix  apt install postfix |

### 方法2：使用sendmail

参考链接：<https://blog.csdn.net/zpf336/article/details/89295474>

1. 安装sendmail

ubuntu中sendmail函数可以很方便的发送邮件，ubuntu sendmail先要安装两个包，再安装一个工具包。

|  |
| --- |
| apt-get install sendmail  apt-get install sendmail-cf  apt-get install mailutils |

1. 配置sendmail

sendmail 默认只会为本机用户发送邮件，只有把它扩展到整个Internet，才会成为真正的邮件服务器。

|  |
| --- |
| #打开sendmail的配置宏文件：/etc/mail/sendmail.mc，  nano /etc/mail/sendmail.mc  #修改如下行  #DAEMON\_OPTIONS(`Family=inet, Name=MTA-v4, Port=smtp, Addr=127.0.0.1')dnl  #修改为  DAEMON\_OPTIONS(`Family=inet, Name=MTA-v4, Port=smtp, Addr=0.0.0.0')dnl  # 0.0.0.0表名可以连接任何服务器  #打开配置信息目录  cd /etc/mail  #做一个备份  mv sendmail.cf sendmail.cf\_bk  #>的左右有空格，提示错误代表没有安装sendmail-cf  m4 sendmail.mc > sendmail.cf |

安装sendmail已经结束，可以配合slurm里的mail参数使用。或者通过

|  |
| --- |
| echo “邮件正文again” | mail -s 邮件主题 邮件地址 |

来进行测试。

## 在控制节点创建多用户，进行用户管理

首先需要在控制节点切换到root用户，创建多个用户。

### 添加用户

|  |
| --- |
| #创建用户  useradd 用户名 -m -s /bin/bash  #创建密码  passwd 用户名  输入密码  #切换到所创建的用户账号，在share文件夹下创建属于自己的文件夹  su 用户名  mkdir /home/share/用户文件夹  #切换回root账户，更改每个用户文件夹的权限为700  exit  chmod 700 /home/share/用户文件夹 |

### 修改用户SSH登录时的默认位置

参考链接：<https://www.jianshu.com/p/88e2f10c6fcd>

为了防止用户在其他文件夹修改配置的情况发生，统一将用户登录后的位置设置为/home/share/用户文件夹 下，此文件夹在控制节点上，但通过NFS使得所有计算节点都可以访问到。

|  |
| --- |
| #执行以下代码，修改passwd配置文件  nano /etc/passwd #找到用户所在行，找到所在行类似下面这种(一般位于最后几行)：  #smy:x:1000:1000:Shelming.Song:/home/smy:/bin/bash  将其中的/home/smy改成你想要修改成的目录（采用绝对路径），如/home/share/用户文件夹，其余内容都别改，保存后关闭。 关闭终端，重新打开，就自动进入你修改的目录了。 |

## 配制开机自启动服务器脚本

由于断电等不确定性因素，服务器可能会关机或者重启，由于munge和slurm都需要手动启动，每次开机后共享文件夹也需要重新挂载，所以需要配置自启动脚本，降低维护成本。

参考链接：<https://blog.csdn.net/ererfei/article/details/98205463>

首先编写自己需要的脚本，其中对于控制节点，需要启动munge和本身以及集群的slurm，不需要挂载共享文件夹。但是对于计算节点，需要启动自己的munge、slurm和挂载共享文件夹。

### 创建需要执行的脚本

使用root权限创建需要执行的脚本slurmstart.sh，这样就会以root权限开机自启动脚本，省去了输入密码的麻烦。

|  |
| --- |
| #创建脚本(位置随意，此处是为了查找方便)  nano /etc/slurmstart.sh |

对于控制节点：

|  |
| --- |
| #!/bin/bash  #启动munge  /etc/init.d/munge start  #启动slurm集群  /opt/slurm/sbin/slurmctld -c  #启动控制节点slurm  /opt/slurm/sbin/slurmd -c |

对于计算节点：

|  |
| --- |
| #!/bin/bash  #启动munge  /etc/init.d/munge start  #启动计算节点slurm  /opt/slurm/sbin/slurmd -c  #重新挂载共享文件夹  mount control:/home/share/ /home/share |

### 将该脚本添加到rc.local中

Ubuntu-18.04以后默认是没有 /etc/rc.local 这个文件的，需要自己创建。

|  |
| --- |
| nano /etc/rc.local  #将以下内容复制进去  #!/bin/sh -e  #  # rc.local  #  # This script is executed at the end of each multiuser runlevel.  # Make sure that the script will "exit 0" on success or any other  # value on error.  #  # In order to enable or disable this script just change the execution  # bits.  #  # By default this script does nothing.  sh /etc/slurmstart.sh  exit 0 |

倒数第二行是指定开机启动后自执行的命令。

|  |
| --- |
| #给rc.local加可执行权限  chmod +x /etc/rc.local |

### 配置开机自启动链接

Ubuntu 16.10开始不再使用initd管理系统，改用systemd，包括用systemctl命令来替换了service和chkconfig的功能。比如以前启动mysql服务用sudo service mysql start，现在用sudo systemctl start mysqld.service。systemd 默认读取 /etc/systemd/system 下的配置文件，该目录下的文件会链接/lib/systemd/system/下的文件。

执行命令ls /lib/systemd/system可以看到很多启动脚本，其中就有我们需要的rc.local.service，打开脚本内容如下

|  |
| --- |
| # SPDX-License-Identifier: LGPL-2.1+  #  # This file is part of systemd.  #  # systemd is free software; you can redistribute it and/or modify it  # under the terms of the GNU Lesser General Public License as published by  # the Free Software Foundation; either version 2.1 of the License, or  # (at your option) any later version.  # This unit gets pulled automatically into multi-user.target by  # systemd-rc-local-generator if /etc/rc.local is executable.  [Unit]  Description=/etc/rc.local Compatibility  Documentation=man:systemd-rc-local-generator(8)  ConditionFileIsExecutable=/etc/rc.local  After=network.target  [Service]  Type=forking  ExecStart=/etc/rc.local start  TimeoutSec=0  RemainAfterExit=yes  GuessMainPID=no |

一般启动文件需要三个组成部分：

* [Unit]段: 启动顺序与依赖关系
* [Service] 段: 启动行为,如何启动，启动类型
* [Install] 段: 定义如何安装这个配置文件，即怎样做到开机启动

上面少了[Install] 段，把下面Install段添上去（使用nano即可）：

|  |
| --- |
| [Install]  WantedBy=multi-user.target  Alias=rc-local.service |

前面我们说 systemd 默认读取 /etc/systemd/system 下的配置文件, 所以还需要在 /etc/systemd/system 目录下创建软链接

|  |
| --- |
| ln -s /lib/systemd/system/rc.local.service /etc/systemd/system/ |

### 启动服务并检查服务状态

检测配置是否成功，可以提前将munge和slurm都关闭，进行测试。

|  |
| --- |
| #授权rc-local.service文件  sudo systemctl enable rc-local  #该处命令是为了在开机状态下检测是否成功，避免重启测试，可以不执行。  sudo systemctl start rc-local.service  #查看测试状态  sudo systemctl status rc-local.service |

## 安装计算环境

为了使得服务器能运行不同语言的的代码，需要安装各种环境，由于在（四、1）中安装munge时已经装了C的运行环境，且Ubuntu系统自带python运行环境，所以这里只给出了Java和matlab的方法。

### Java运行环境的安装

参考链接：<https://cloud.tencent.com/developer/article/1162527>

|  |
| --- |
| #先更新本地软件包：  apt update  #检查是否已安装Java：  java -version  #如果当前未安装Java，您将看到以下输出：  #Command 'java' not found, but can be installed with:  #apt install default-jre  #apt install openjdk-11-jre-headless  #apt install openjdk-8-jre-headless  #apt install openjdk-9-jre-headless  #执行以下命令安装OpenJDK：  apt install default-jre  #验证是否安装完成：  java -version  #您将看到以下输出：  openjdk version "10.0.1" 2018-04-17  OpenJDK Runtime Environment (build 10.0.1+10-Ubuntu-3ubuntu1)  OpenJDK 64-Bit Server VM (build 10.0.1+10-Ubuntu-3ubuntu1, mixed mode)  #除了JRE之外，还需要Java Development Kit（JDK）才能编译和运行某些特定的基于Java的软件。要安装JDK，请执行以下命令，该命令还将安装JRE：  apt install default-jdk  #通过检查Java编译器的javac版本，来验证是否已安装JDK ：  javac -version  #将看到以下输出：  javac 10.0.1 |

### Matlab安装

由于正版matlab的licenses文件需要绑定用户，且只能绑定一个用户，无法实现对所有普通用户的使用，所以安装了破解版的matlab，下载来源为学校睿思：<http://rs.xidian.edu.cn/forum.php?mod=viewthread&tid=1076337>。

具体安装方法参考软件包中的readme说明文档，且由于破解版本较多，教程无太大参考价值，此处不做赘述。重点介绍可以对 matlab 进行支持，显示 Matlab 的图标和提供一些其他的功能的matlab-support工具的安装。

|  |
| --- |
| #安装 matlab-support  apt install matlab-support  #安装过程中需要输入matlab安装路径例如  /usr/local/polyspace/R2020b  #同时需要指定使用用户，指定  nobody  #最后还需要输入一次matlab安装路径  /usr/local/polyspace/R2020b |

至此，服务器所有配置已经结束。完结撒花~~~~。