在线判题系统后台评测模块环境配置

注:本次环境配置测试均在Ubuntu 14.04 32位英文版上进行。

消息系统RabbitMQ的安装与使用(仅架设RabbitMQ的服务器需要配置)

1. 安装

在Ubuntu 14.04下可以直接通过apt-get安装:

sudo apt-get install rabbitmq-server



安装好后，rabbitmq服务就已经启动好了。如果rabbitmq服务没有启动,可用如下命令启动:

sudo rabbitmq-server start

如果想关闭rabbitmq服务,用以下命令:

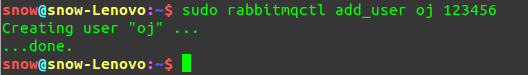
sudo rabbitmq-server stop

查看rabbitmq服务状态: sudo rabbitmqctl status

1. 添加用户

为了后台评测模块运行的安全性,我们需要为程序新建一个用户:

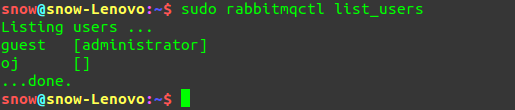
sudo rabbitmqctl add\_user oj 123456



这样我们就新建了一个可以连接到rabbitmq的用户，用户名是oj,密码是123456.

可以通过 sudo rabbitmqctl change\_password oj 666666修改密码.

我们可以用 sudo rabbitmqctl list\_users 查看到我们新添加的用户:



可以看到添加用户成功了，但不是administrator角色，这里我们也可以将oj用户设置为administrator角色.

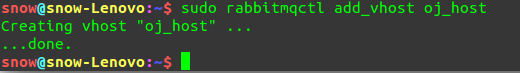
sudo rabbitmqctl set\_user\_tags oj administrator

如果不再需要该用户,可以通过 sudo rabbitmqctl delete\_user oj删除该用户.

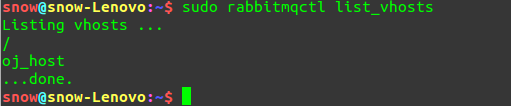
1. 添加虚拟主机(virtual host)

同样为了后台评测模块运行的安全性,我们还要新建一个virtual host:

sudo rabbitmqctl add\_vhost oj\_host



我们可以通过 sudo rabbitmqctl list\_vhosts命令查看到我们添加的虚拟主机.



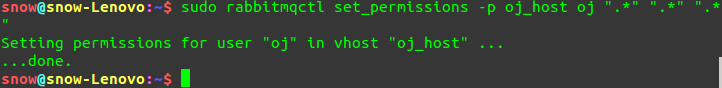
可以看到系统现在有两个virtual host，一个是系统默认的 '/', 还有一个就 是我们新建的 oj\_host。

如果不再需要该virtual host,可以通过 sudo rabbitmqctl delete\_vhost oj\_host 删除该virtual host.

1. 为用户分配访问权限

为了能让我们新建的用户oj可以通过外网使用这个用户名和密码访问虚拟主机oj\_host,我们还要给它分配访问权限。

sudo rabbitmqctl set\_permissions -p oj\_host oj ".\*" ".\*" ".\*"



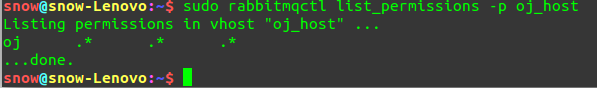
如此用户名为oj的用户就可以访问vitrual host为oj\_host的资源了，并且具备配置、读、写的权限。

若权限分配错误,可使用如下命令清除用户权限:

sudo rabbitmqctl clear\_permissions -p oj\_host oj

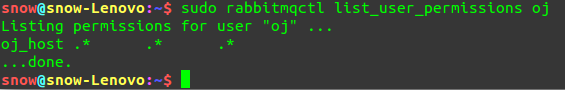
查看虚拟主机上的所有权限:

sudo rabbitmqctl list\_permissions -p oj\_host



也可查看用户oj的权限:

sudo rabbitmqctl list\_user\_permissions oj



远程连接SSH的安装与配置(仅后台评测系统的分布式控制中心需要配置)

1. 安装

Ubuntu 下安装 OpenSSH Server 是一件无比轻松的事情，需要的命令只有一条：

sudo apt-get install openssh-server

一路回车, SSH服务就安装成功啦.

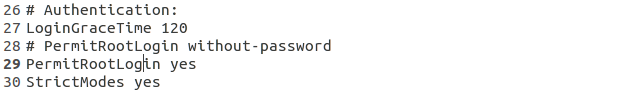
我们可以通过 sudo ps -e | grep ssh 来查看ssh服务是否启动:



如有sshd, 说明ssh服务已经启动，如果没有启动，可以通过 sudo service ssh start 来启动ssh服务.

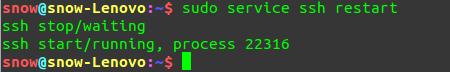
1. 配置SSH服务

对于设置有密码的root用户, 刚安装完成的ssh服务是不允许其登录的, 我们可以通过 sudo gedit /etc/ssh/sshd\_config来修改配置文件.



将配置文件中的"PermitRootLogin without-password"加一个"#"号,将其注释掉,再增加一句"PermitRootLogin yes",然后保存即修改成功。

然后通过 sudo service ssh restart 来重启服务:



这样就可以通过ssh用现有的用户名和密码登录系统了.

Python虚拟环境的使用(可选)

虚拟环境是 Python 解释器的一个私有副本,在这个环境中你可以安装私有包,而且不会影响系统中安装的全局 Python 解释器。

虚拟环境非常有用,可以在系统的 Python 解释器中避免包的混乱和版本的冲突。为每个程序单独创建虚拟环境可以保证程序只能访问虚拟环境中的包,从而保持全局解释器的干净整洁,使其只作为创建(更多)虚拟环境的源。使用虚拟环境还有个好处,那就是不需要管理员权限。

虚拟环境使用第三方实用工具 virtualenv 创建。输入以下命令可以检查系统是否安装了virtualenv :



如果结果显示错误,你就需要安装这个工具。

Ubuntu 用户可以使用下述命令安装它:



接下来我们要在应用程序文件夹oj下使用 virtualenv 命令创建 Python 虚拟环境.这个命令只有一个必需的参数,即虚拟环境的名字。创建虚拟环境后,当前文件夹中会出现一个子文件夹,名字就是上述命令中指定的参数,与虚拟环境相关的文件都保存在这个子文件夹中。按照惯例,一般虚拟环境会被命名为 venv :



现在,oj文件夹中就有了一个名为 venv 的子文件夹,它保存一个全新的虚拟环境,其中有一个私有的 Python 解释器。在使用这个虚拟环境之前,你需要先将其“激活”。可以通过下面的命令激活这个虚拟环境:



虚拟环境被激活后,其中 Python 解释器的路径就被添加进 PATH 中,但这种改变不是永久性的,它只会影响当前的命令行会话。为了提醒你已经激活了虚拟环境,激活虚拟环境的命令会修改命令行提示符,加入环境名:



当虚拟环境中的工作完成后,如果你想回到全局 Python 解释器中,可以在命令行提示符下输入 deactivate 。

需求文件的管理与使用(后台评测系统的分布式控制中心与评判节点均需配置)

在我们的在线判题系统的后台评测模块中,除了python的标准库,我们还使用了一些第三方扩展包.为了方便的管理这些第三方包,我们已经在程序中包含了一个 requirements.txt 文件,用于记录所有依赖包及其精确的版本号。如果要在另一台电脑上(例如部署程序时使用的电脑)重新生成虚拟环境,这个文件的重要性就体现出来了。 pip 可以使用如下命令自动生成这个文件:



安装或升级包后,最好更新这个文件。如果你要创建这个虚拟环境的完全副本,可以创建一个新的虚拟环境,并在其上运行以下命令:



编译安装paramiko(仅分布式评测节点需要安装)

paramiko是用python语言写的一个模块，遵循SSH2协议，支持以加密和认证的方式，进行远程服务器的连接。

由于使用的是python这样的能够跨平台运行的语言，所以所有python支持的平台，如Linux, Solaris, BSD, MacOS X, Windows等，paramiko都可以支持，因此，如果需要使用SSH从一个平台连接到另外一个平台，进行一系列的操作时，paramiko是最佳工具之一。

使用paramiko仅需要在本地上安装相应的软件（python以及PyCrypto），对远程服务器没有配置要求.

安装paramiko有两个先决条件，python和另外一个名为PyCrypto的模块。在此我们仅需下载安装paramiko, 在安装过程中安装程序会自动检查包依赖问题并帮我们解决.首先我们需要到github上下载paramiko的最新稳定版:

<https://github.com/paramiko/paramiko>

下载完后解压缩,在虚拟环境中进入解压缩后的目录,在模块的根目录下运行：

python setup.py build  
  
 python setup.py install

以上两条命令即可安装.

编译安装lorun(仅分布式评测节点需要安装)

lorun是github上的一个开源项目，这是用C语言写的一个python扩展模块，让程序在一个类似沙盒的环境下执行，然后精准的获取程序的执行时间和内存，还能对程序进行限制，限制程序的系统调用。我们利用它来测量程序的执行时间和内存占用情况. 为此我们先要到项目地址 <https://github.com/lodevil/Lo-runner> 将其下载下来.

然后解压缩,在虚拟环境中进入解压缩后的目录,在模块的根目录下运行：

python setup.py install

即可.