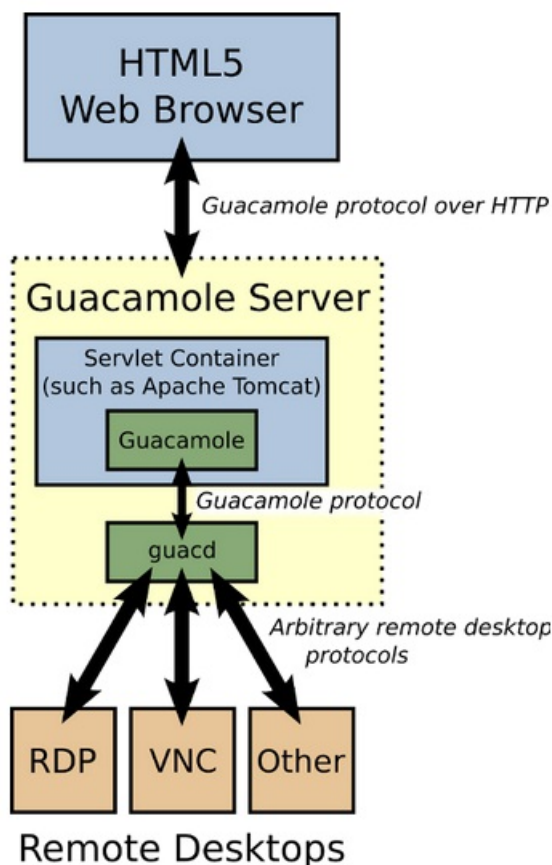


一. 介绍

最近在看一些虚拟化解决方案方面的工作，每次都被开网页的虚拟机界面提示安装JRE搞的精神都不好了，因此希望能在浏览器上远程操作虚拟机，不想安装任何插件。

通过搜索发现了Guacamole，一个提供远程桌面的解决方案的开源项目，通过浏览器就能操作虚拟机，适用于Chrome、Firefox、IE9+等浏览器（浏览器需要支持HTML5），由于使用HTML5，Guacamole只要在一个服务器安装成功，你访问你的桌面就是访问一个web浏览器。它目前是一个开源的项目，我们可以对Guacamole做修改，适配成我们需要的项目，比如做成自动登录的，加上项目权限验证等。

Guacamole—基于HTML5的远程桌面控制框架，性能较高，官方说接近于原生的VNC，Guacamole不是一个独立的Web应用程序，而是由许多部件组成的。Web应用程序实际上是整个项目里最小最轻量的，大部分的功能依靠Guacamole的底层组件来完成，整个项目的构建如下图：



从上图可以看出，guacamole结构上分为4层，建议先阅读下<http://guac-dev.org/doc/gug/guacamole-architecture.html>和<http://guac-dev.org/doc/gug/guacamole-protocol.html>，就可以对Guacamole的架构和协议有个基本的认识。

guacamole程序的流程如下：用户通过浏览器连接到Guacamole的服务端，Guacamole的客户端是用javascript编写的，Guacamole Server通过Web容器（比如Tomcat）把服务提供给用户。一旦加载，客户端通过http承载着Guacamole自己的定义的协议与服务端通信。部署在Guacamole Server这边的Web应用程序，解析到的Guacamole protocol，就传给Guacamole的代理guacd，这个代理（guacd）实际上就是解析Guacamole protocol，替用户连接到远程机器，Guacamole protocol协议本身以及guacd的存在，实现了协议的透明：Guacamole客户端（浏览器运行的JS）和Web应用程序，都不需要知道远程桌面具体用哪个协议（VNC,RDP etc），分为4个大的部分：

①. JS (WebSocket/xmlhttprequest + canvas), 普通的用户看到的部分, 使用的HTML5与后台进行交互

②. JavaServlet: 处理与用户的交互, 将页面上的操作请求处理下, 再直接与下层的guacd来交互

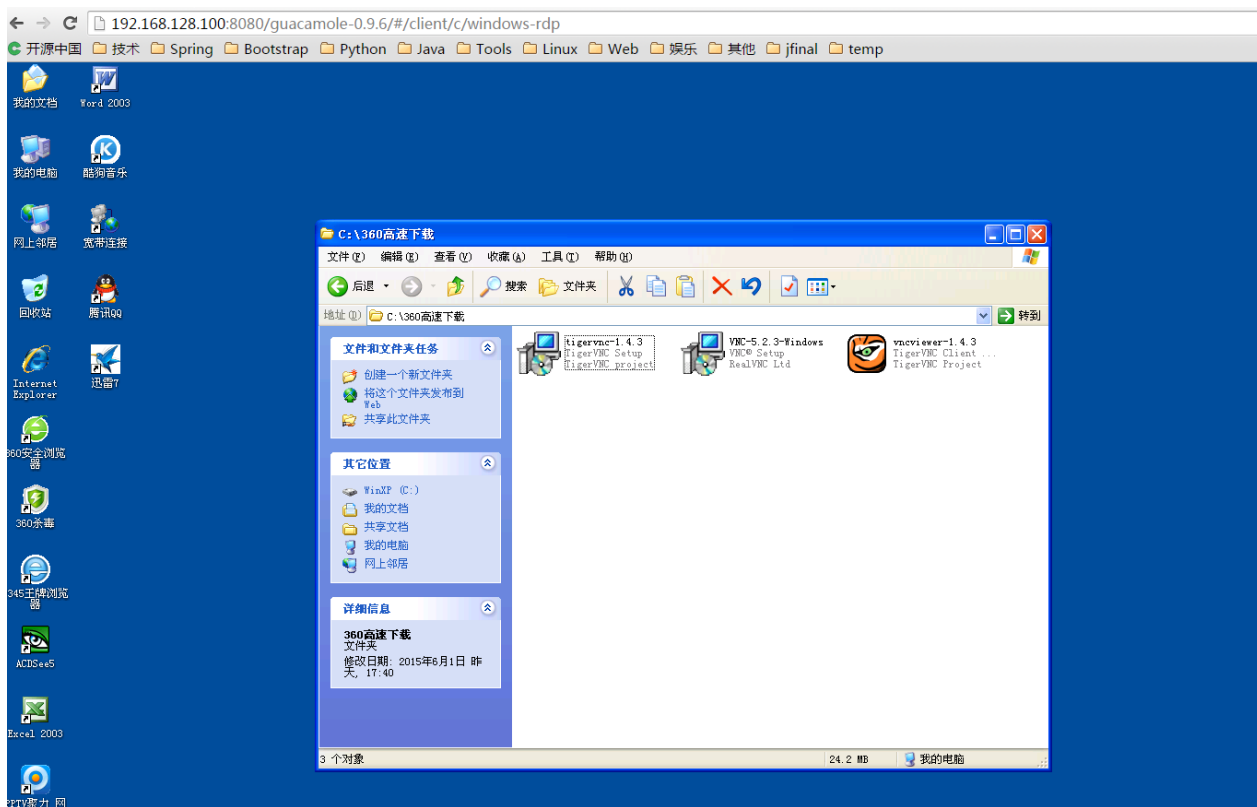
③. guacd, 底层的daemon, 封装了各种RDP协议的中间层, 如VNC等。

④. libfreerdp, libssh等, 各种通信协议, 会被封装成Guacamole的协议

而关于Guacamole的协议, JS与JavaServlet, Servlet与guacd交互, 都是采用这种格式。这是网上有人从guacd中抓下来的部分日志, 与原文一致

```
4.sync,13.1386052271656;0,1.0,1.7,2.16;1,2.14,1.0,1.0,1.0,1.0,3.255;4.rect,1.0,3.160,3.285,2.10,2.19;5.cfill,2.14,1.0,3.153,3.153,3.255;4.rect,2,-1,1.0,1.0,3.520,2.19;5.cfill,1.6,2,-1,3.103,3.255,3.103,3.255;4.rect,2,-2,1.0,1.0,3.520,2.19;5.cfill,2.14,2,-2,1.0,1.0,3.520,2.19,2.14,2,-2,1.0,1.0,3.255;4.copy,2,-2,2.10,1.0,2.10,2.19,2.14,1.0,1.0,3.285;4.copy,2,-2,2.20,1.0,2.10,2.19,2.14,1.0,2.10,3.285;4.rect,2,-1,3.520,1.0,2.10,2.19;5.cfill,1.2,2,-1,1.0,1.0,1.0,3.255;3.png,2.14,2,-1,3.520,1.0,380.iVBORw0KGgoAAAANSUHEUgAAAAcAAAAATCAyAAACp65uAAAABmJLR0QA/vD/AP+gvaetAAAAOE1EQWQkeXQ0OodURSA4e/eHVjaIjWgWNI167AZV5BnZekqLBMmVboM BMQXgjYuwS3YpkmC+IqEuTzJHMkKxL89H+fa4T8Wgm1USpHOM:DF14dn1ljCc95yI9jN3QneEBWRBvU28EBCogQhAG06gm+tyRRzPCWpJsj0IVFoJQ3KqC617ova9gJ3Reg3C7h03U3sU+Bj8XF1WpSNJhMzW35+YZ JmPj+eU4M+vDC7LF5ethGH6twXof+IySaFYR5VF9qVR/WbYxShq+uIF9Q20D0R0ZK/+VAAAAABJRUSerkJggg==
```

远程访问Windows如下图所示:



二. 部署环境和优势

1. Guacamole完全基于浏览器, 不必安装JRE、RDP、VNV等客户端插件, 不需要考虑客户的机器环境

其实远程桌面本身有很多实现和协议, 可以通过Applet、VNC Viewer、RDP等进行管理, 但是都需要客户安装一些插件或者软件, 还要区分用户是32位或者64位系统等等, 比较繁琐,

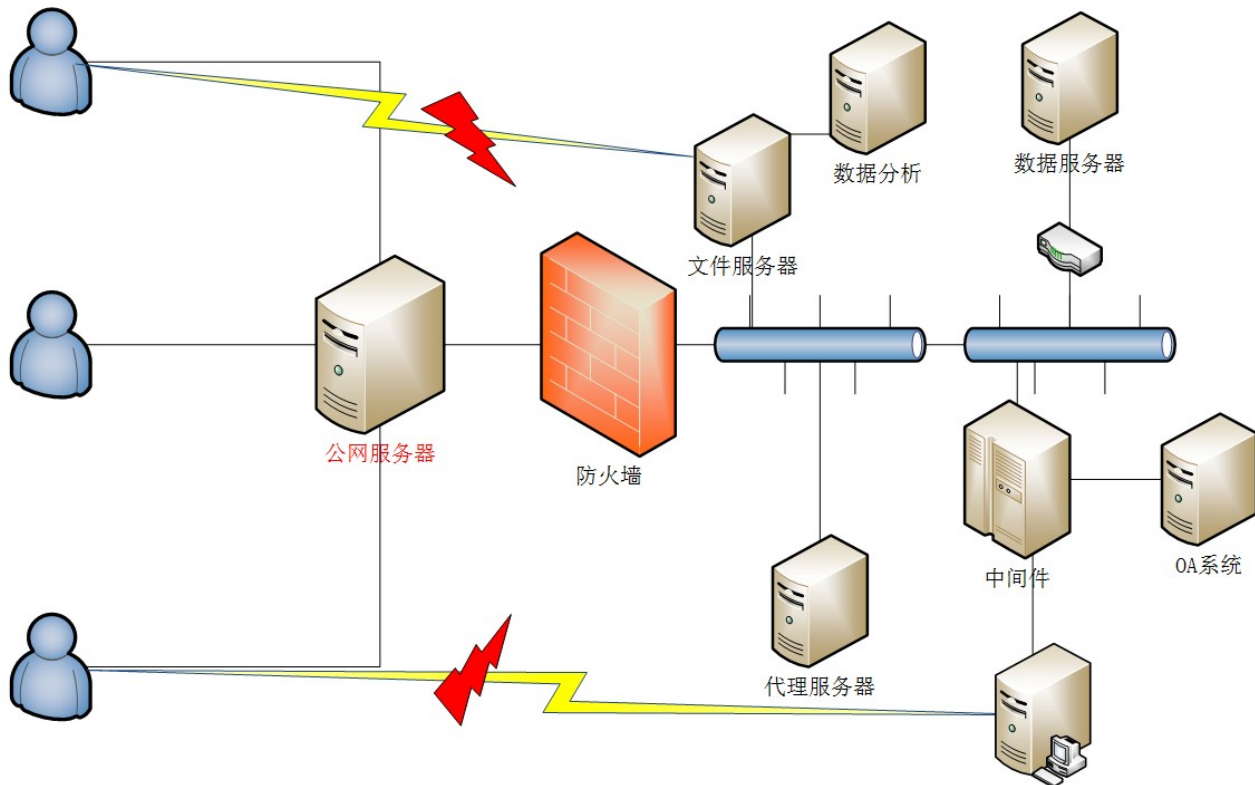
而有的客户只是临时使用一下, 也不想为自己的机器装一堆软件。

2. Guacamole可以通过一台授权机器管理多台服务器, 免于多个管理端登录。

一些复杂的网络环境, 可能有很多台服务器, 比如防火墙、路由器、文件服务器、路由服务器等等, 用户如果想要管理, 需要多次打开远程工具, 连接不同的设备进行管理,

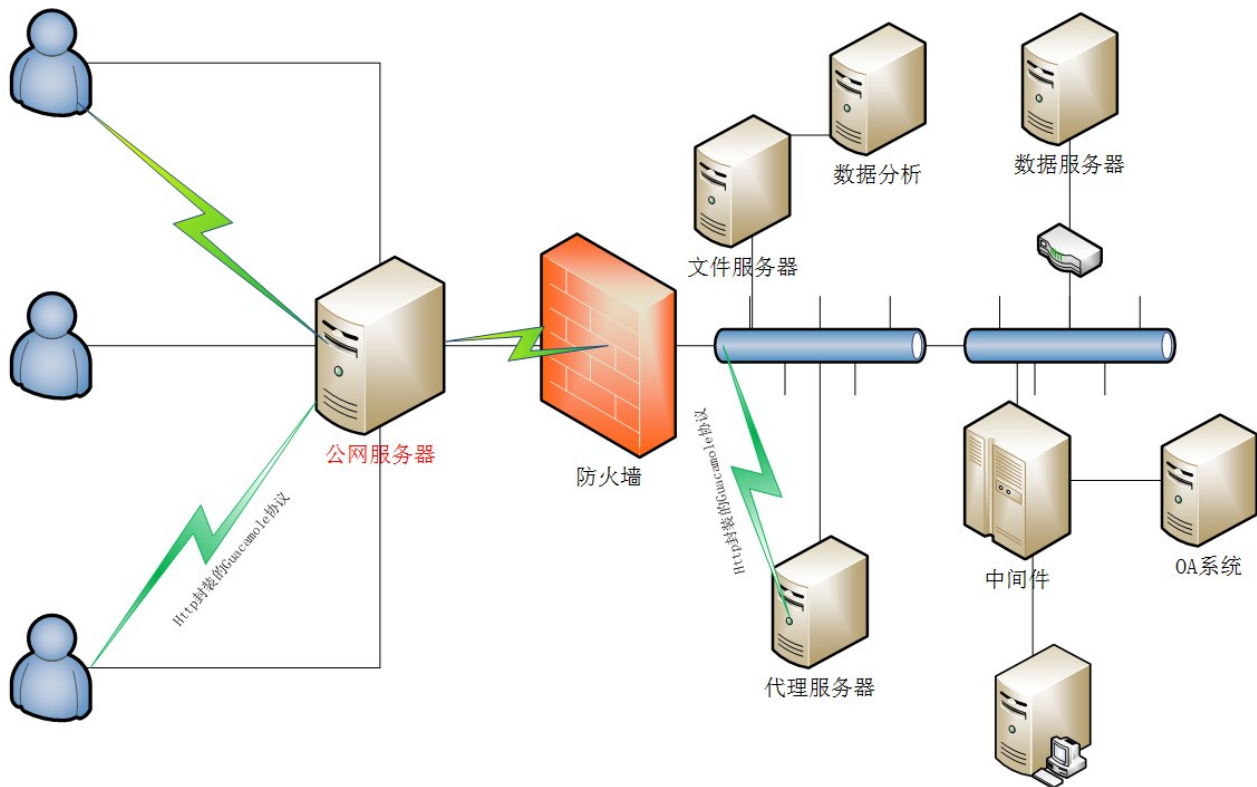
繁琐而且容易出现密码忘记等问题

3. Guacamole可以穿透防火墙，通过一台服务器管理内容的其它不对外提供网络服务的设备，如下图所示:



除了挂在公网上的服务器，其它设备是没办法直接连接公网的，这时候如果一台设备出现问题，我们是没办法远程调试的，但是通过Guacamole，我们可以管理内网

的多台机器，只要保证Guacamole服务器和内网是可以进行数据通信的，如下图所示:



4. 在云项目中，我们需要面对的就是一个个虚拟机器，我们没办法通过真实的设备查看，只能通过虚拟桌面，而目前远程虚拟桌面大部分都是通过安装各种客户端，

而这里我们可以通过数据库或者XML文件，配置KVM虚拟机的端口，直接通过网页管理虚拟机，设置都不需要再次进行验证，直接登录

5. Guacamole—基于HTML5的远程桌面控制框架，性能较高，官方说接近于原生的VNC，对于一些临时性管理和查看任务足够了。

三. 安装

1. 安装环境说明

我安装是基于CentOS6.5 X86_64位进行的，其它版本也相差不大，主要区别也就是编译server的一些依赖库不同.下面的图是安装所需要的所有文件:

```
[root@zeyu guacamole]# ll
总用量 153500
-rw-r--r-- 1 root root 8042383 6月 3 05:40 apache-maven-3.3.3-bin.tar.gz
-rw-r--r-- 1 root root 8824528 6月 3 05:40 apache-tomcat-7.0.62.tar.gz
drwxr-xr-x 9 root root 4096 6月 3 06:05 guacamole-client-0.9.6
-rw-r--r-- 1 root root 1506395 6月 3 05:42 guacamole-client-0.9.6.tar.gz
drwxrwxr-x 7 1000 1000 4096 6月 3 06:17 guacamole-server-0.9.6
-rw-r--r-- 1 root root 633621 6月 3 05:42 guacamole-server-0.9.6.tar.gz
-rw-r--r-- 1 root root 138082565 6月 3 05:36 jdk-7u79-linux-x64.rpm
-rw-r--r-- 1 root root 22528 6月 3 06:01 libtelnet-0.20-2.el6.x86_64.rpm
-rw-r--r-- 1 root root 25756 6月 3 06:01 libtelnet-devel-0.20-2.el6.x86_64.rpm
drwxr-xr-x 6 root root 4096 6月 3 05:40 maven
drwxr-xr-x 2 root root 4096 6月 3 05:42 mysql
drwxr-xr-x 2 root root 4096 6月 3 05:42 noauth
drwxr-xr-x 10 root root 4096 6月 3 06:18 tomcat
drwxr-xr-x 2 root root 4096 6月 3 05:42 xml
```

其中Maven是client端的编译工具，JDK和Tomcat是运行Client端的环境，mysql、noauth和xml分别是不同授权模式下的配置文件；

而guacamole-server-0.9.6.tar.gz和guacamole-client-0.9.6.tar.gz就是源码了。

2. 安装Server

开始编译Server端了：

首先解压压缩包：

```
[root@zeyu guacamole]# tar -zxf guacamole-server-0.9.6.tar.gz
```

进入解压缩后的目录，以后对server端的操作，我们都基于这个目录：

```
[root@zeyu guacamole-client-0.9.6]# cd ../guacamole-server-0.9.6
[root@zeyu guacamole-server-0.9.6]#
```

首先我们安装编译必须的gcc make等工具：

```
[root@zeyu guacamole-server-0.9.6]# yum install libpng* freerdp freerdp-devel libvncserver libvncserver-devel cairo-devel gcc* gcc++ make automake libtool
```

安装所有依赖库，主要是RDP、VNC、SSH、TELENT等，不需要那个协议可以不安装：

Library name	Features
FreeRDP	<p>FreeRDP is required for RDP support. If you do not wish to build RDP support, this library is not needed.</p> <p>Debian / Ubuntu package libfreerdp-dev</p> <p>Fedora / CentOS / RHEL package freerdp-devel</p>
Pango	<p>Pango is a text layout library which Guacamole's SSH and telnet support uses to render text. If you do not wish to build SSH or telnet support, this library is not needed.</p> <p>Debian / Ubuntu package libpango1.0-dev</p> <p>Fedora / CentOS / RHEL package pango-devel</p>
libssh2	<p>libssh2 is required for SSH support. If you do not wish to build SSH support, this library is not needed.</p> <p>Debian / Ubuntu package libssh2-1-dev</p> <p>Fedora / CentOS / RHEL package libssh2-devel</p>
libtelnet	<p>libtelnet is required for telnet support. If you do not wish to build telnet support, this library is not needed.</p> <p>Debian / Ubuntu package libtelnet-dev</p> <p>Fedora / CentOS / RHEL package libtelnet-devel</p>
libVNCServer	<p>libVNCServer provides libvncclient, which is required for VNC support. If you do not wish to build VNC support, this library is not needed.</p> <p>Debian / Ubuntu package libvncserver-dev</p> <p>Fedora / CentOS / RHEL package libvncserver-devel</p>
PulseAudio	<p>PulseAudio provides libpulse, which is used by Guacamole's VNC support to provide experimental audio support. If you are not going to be using the experimental audio support for VNC, you do not need this library.</p> <p>Debian / Ubuntu package libpulse-dev</p> <p>Fedora / CentOS / RHEL package pulseaudio-libs-devel</p>
OpenSSL	<p>OpenSSL provides support for SSL and TLS - two common encryption schemes that make up the majority of encrypted web traffic.</p> <p>If you have libssl installed, guacd will be built with SSL support, allowing communication between the web application and guacd to be encrypted. This library is also required for SSH support for the sake of manipulating public/private keys.</p> <p>Without SSL support, there will be no option to encrypt communication to guacd, and support for SSH cannot be built.</p> <p>Debian / Ubuntu package libssl-dev</p> <p>Fedora / CentOS / RHEL package openssl-devel</p>
libvorbis	<p>libvorbis provides support for Ogg Vorbis - a free and open standard for sound compression. If installed, libguac will be built with support for Ogg Vorbis, and protocols supporting audio will use Ogg Vorbis compression when possible.</p> <p>Otherwise, sound will only be encoded as WAV (uncompressed), and will only be available if your browser also supports WAV.</p> <p>Debian / Ubuntu package libvorbis-dev</p> <p>Fedora / CentOS / RHEL package libvorbis-devel</p>

```
[root@zeyu guacamole-server-0.9.6]# yum install uuid uuid-devel cairo-devel libpng-devel freerdp-devel pango-devel libssh2-devel libvncserver-devel
```

这里不同的CenOS可能有的装不上:


```
[root@zeyu guacamole-server-0.9.6]# yum install libtelnet-devel pulseaudio-libs-devel openssl-devel libvorbis-devel
```

Telnet协议CentOS已经不能在线安装了，因为安全性太差，这里为了测试，我去RedHat上下的安装包：

```
[root@zeyu guacamole-server-0.9.6]# rpm -vih ../libtelnet-*
```

安装完成后可以进行Server端的编译了：

```
[root@zeyu guacamole-server-0.9.6]# ./configure --with-init-dir=/etc/init.d
```

configure完成结果是这样，如果你有的协议不需要，support那里可能是no：

```
-----  
guacamole-server version 0.9.6  
-----
```

Library status:

```
freerdp ..... yes  
pango ..... yes  
libssh2 ..... yes  
libssl ..... yes  
libtelnet ..... yes  
libVNCServer ..... yes  
libvorbis ..... yes  
libpulse ..... yes
```

Protocol support:

```
RDP ..... yes  
SSH ..... yes  
Telnet .... yes  
VNC ..... yes
```

Init scripts: /etc/init.d

Type "make" to compile guacamole-server.

```
[root@zeyu guacamole-server-0.9.6]# make & make install
```

运行一下ldconfig命令，使配置生效：

```
[root@zeyu guacamole-server-0.9.6]# ldconfig
```

通过脚本激动Server端，如果出现SUCCESS则是成功了

```
[root@zeyu guacamole-server-0.9.6]# /etc/init.d/guacd start
Starting guacd: SUCCESS
```

Server端至此编译和安装完成.

3. 安装Client

首先安装JDK:

```
[root@zeyu guacamole]# rpm -vih jdk-7u79-linux-x64.rpm
```

安装完成后使用命令验证:

```
[root@zeyu guacamole]# java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
```

然后是Tomcat和Maven, 它们都是绿色版的, 只需要解压缩就可以了:

```
[root@zeyu guacamole]# tar -zxf apache-maven-3.3.3-bin.tar.gz
```

```
[root@zeyu guacamole]# tar -zxf apache-tomcat-7.0.62.tar.gz
```

为了操作方便, 简化一下目录名称:

```
[root@zeyu guacamole]# mv apache-maven-3.3.3 maven
```

```
[root@zeyu guacamole]# mv apache-tomcat-7.0.62 tomcat
```

启动tomcat, 通过进程或者端口查看是否成功:

```
[root@zeyu guacamole]# tomcat/bin/startup.sh
```

```
[root@zeyu guacamole]# ps -eaf|grep tomcat
root      9711      1  0 07:48 pts/0    00:01:59 /usr/java/jdk1.7.0_79/jre/bin/java -Djava.util.logging.config.file=/guacamole/tomcat/conf/
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/guacamole/tomcat/endorsed -classpath /guacamole/tor
jar:/guacamole/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/guacamole/tomcat -Dcatalina.home=/guacamole/tomcat -Djava.io.tmpdir=/guacamole/
che.catalina.startup.Bootstrap start
root      9883  9556  4 13:59 pts/0    00:00:00 grep tomcat
[root@zeyu guacamole]#
```

PS: 这里修改了一下环境变量, 因为我在后面启动client端的时候发现总是启动失败, 查看代码发现这个:


```
*/
public static File getDirectory() {

    // Attempt to find Guacamole home
    File guacHome;

    // Use system property by default
    String desiredDir = System.getProperty("guacamole.home");

    // Failing that, try the GUACAMOLE_HOME environment variable
    if (desiredDir == null) desiredDir = System.getenv("GUACAMOLE_HOME");

    // If successful, use explicitly specified directory
    if (desiredDir != null)
        guacHome = new File(desiredDir);

    // If not explicitly specified, use ~/.guacamole
    else
        guacHome = new File(System.getProperty("user.home"), ".guacamole");

    // Return discovered directory
    return guacHome;

}
```

它的代码里读取配置文件的方法有问题，除了改代码，只能先设置环境变量了：

```
[root@zeyu guacamole]# vi /etc/profile
```

```
export JAVA_HOME="/usr/java/jdk1.7.0_79"
export JRE_HOME="$JAVA_HOME/jre"
export CLASSPATH="$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH"
export PATH="$JAVA_HOME/bin:$JRE_HOME/bin:$PATH"
export TOMCAT_HOME="/guacamole/tomcat"
export CATALINA_HOME="/guacamole/tomcat"

-- INSERT --
```

使环境变量生效：

```
[root@zeyu guacamole]# source /etc/profile
```

开始编译client端了：

首先解压压缩包：

```
[root@zeyu guacamole]# tar -zxf guacamole-client-0.9.6.tar.gz
```

直接调用mvn命令，这里没设置maven环境变量，只能使用全路径：

```
[root@zeyu guacamole]# ../maven/bin/mvn package
```

为了提高效率，可以跳过测试进行编译

```
[root@zeyu guacamole]# ../maven/bin/mvn package -DskipTests
```

已经开始编译了，扫描工程，共有11个子工程，其中postgresql、ldap工程我用不到，在导入工程时删除了：

```
[INFO] Reactor Build Order:
[INFO]
[INFO] guacamole-common
[INFO] guacamole-ext
[INFO] guacamole-common-js
[INFO] guacamole
[INFO] guacamole-auth-jdbc
[INFO] guacamole-auth-jdbc-base
[INFO] guacamole-auth-jdbc-mysql
[INFO] guacamole-auth-jdbc-postgresql
[INFO] guacamole-auth-ldap
[INFO] guacamole-auth-noauth
[INFO] guacamole-client
[INFO]
[INFO] -----
[INFO] Building guacamole-common 0.9.6
[INFO] -----
```

编译完成的结果如下图所示：

```
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] guacamole-common ..... SUCCESS [ 40.151 s]
[INFO] guacamole-ext ..... SUCCESS [ 9.261 s]
[INFO] guacamole-common-js ..... SUCCESS [ 3.076 s]
[INFO] guacamole ..... SUCCESS [ 15.712 s]
[INFO] guacamole-auth-jdbc ..... SUCCESS [ 2.645 s]
[INFO] guacamole-auth-jdbc-base ..... SUCCESS [ 1.328 s]
[INFO] guacamole-auth-jdbc-mysql ..... SUCCESS [ 2.596 s]
[INFO] guacamole-auth-jdbc-postgresql ..... SUCCESS [ 1.963 s]
[INFO] guacamole-auth-ldap ..... SUCCESS [ 0.633 s]
[INFO] guacamole-auth-noauth ..... SUCCESS [ 0.326 s]
[INFO] guacamole-client ..... SUCCESS [ 4.444 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:23 min
[INFO] Finished at: 2015-06-03T14:06:01+08:00
[INFO] Final Memory: 40M/95M
[INFO] -----
[root@zeyu guacamole-client-0.9.6]#
```

编译出来的war包，在这个目录下，可以在tomcat中直接部署：

```
[root@zeyu guacamole-client-0.9.6]# ll guacamole/target/
总用量 6664
drwxr-xr-x  4 root root    4096 6月   3 06:03 classes
drwxr-xr-x  3 root root    4096 6月   3 06:03 generated-sources
drwxr-xr-x 11 root root    4096 6月   3 06:03 guacamole-0.9.6
-rw-r--r--  1 root root 6798647 6月   3 14:05 guacamole-0.9.6.war
drwxr-xr-x  2 root root    4096 6月   3 06:03 maven-archiver
drwxr-xr-x  3 root root    4096 6月   3 06:03 maven-status
drwxr-xr-x  3 root root    4096 6月   3 06:03 war
[root@zeyu guacamole-client-0.9.6]#
```

如果你想在eclipse修改代码，可以通过maven命令生成eclipse工程

```
ix>../maven/bin/mvn eclipse:eclipse
```

工程结构如下（删除了ldap和postgresql）：

- ▷ guacamole
- ▷ guacamole-auth-jdbc-base
- ▷ guacamole-auth-jdbc-mysql
- ▷ guacamole-auth-noauth
- ▷ guacamole-common
- ▲ guacamole-ext
 - ▷ src/test/java
 - ▷ src/main/java
 - ▷ src/main/resources
 - ▷ Referenced Libraries
 - ▷ JRE System Library [jre7]
 - ▷ src
 - ▷ target
 - ChangeLog
 - LICENSE
 - pom.xml
 - README

最后我们把编译好的war包拷贝到tomcat的webapps目录下，建立配置文件的文件夹

```
[root@zeyu guacamole]# mkdir tomcat/.guacamole
```

```
[root@zeyu guacamole]# cp /guacamole/guacamole-client-0.9.6/guacamole/target/guacamole-0.9.6.war webapps/
```

最后启动tomcat，安装工作就完成了。

```
[root@zeyu guacamole]# tomcat/bin/startup.sh
```

```
[root@zeyu guacamole]# tail -f tomcat/logs/catalina.out
六月 03, 2015 7:48:55 上午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-bio-8009"]
六月 03, 2015 7:48:55 上午 org.apache.catalina.startup.Catalina start
信息: Server startup in 7751 ms
```

四. 测试

Guacamole提供了各种模式的登录和权限机制，这里我只测试我可能会用到的几种:XML、NoAuth、JDBC-Mysql,需要的配置文件如下:

```
[root@zeyu guacamole]# ll xml
总用量 8
-rw-r--r-- 1 root root 327 6月 3 05:42 guacamole.properties
-rw-r--r-- 1 root root 2309 6月 3 05:42 user-mapping.xml
[root@zeyu guacamole]# ll noauth/
总用量 16
-rw-r--r-- 1 root root 6775 6月 3 05:42 guacamole-auth-noauth-0.9.6.jar
-rw-r--r-- 1 root root 342 6月 3 05:42 guacamole.properties
-rw-r--r-- 1 root root 492 6月 3 05:42 noauth-config.xml
[root@zeyu guacamole]# ll mysql/
总用量 1896
-rw-r--r-- 1 root root 7580 6月 3 05:42 001-create-schema.sql
-rw-r--r-- 1 root root 1959 6月 3 05:42 002-create-admin-user.sql
-rw-r--r-- 1 root root 99945 6月 3 05:42 guacamole-auth-jdbc-base-0.9.6.jar
-rw-r--r-- 1 root root 27284 6月 3 05:42 guacamole-auth-jdbc-mysql-0.9.6.jar
-rw-r--r-- 1 root root 333 6月 3 05:42 guacamole.properties
-rw-r--r-- 1 root root 34471 6月 3 05:42 guice-multibindings-3.0.jar
-rw-r--r-- 1 root root 714073 6月 3 05:42 mybatis-3.2.8.jar
-rw-r--r-- 1 root root 72415 6月 3 05:42 mybatis-guice-3.6.jar
-rw-r--r-- 1 root root 964879 6月 3 05:42 mysql-connector-java-5.1.31.jar
[root@zeyu guacamole]#
```

1. XML模式测试

先拷贝配置文件:

```
[root@zeyu guacamole]# cp xml/* tomcat/.guacamole/
```

修改guacamole.properties如下所示:

```
[root@zeyu guacamole]# vi xml/guacamole.properties
```

```
# Hostname and port of guacamole proxy
guacd-hostname: localhost
guacd-port: 4822

# Auth provider class (authenticates user/pass combination, needed if using the provided login screen)
auth-provider: net.sourceforge.guacamole.net.basic.BasicFileAuthenticationProvider
basic-user-mapping: /etc/guacamole/user-mapping.xml
```

添加xml/user-mapping.xml文件的选择，里面都是我们需要监控的机器:

```
[root@zeyu guacamole]# vi xml/user-mapping.xml
```

用户名、密码可以有多个，可以明文也可以md5加密:

```
<!-- Per-user authentication and config information -->
<authorize username="USERNAME" password="PASSWORD">
  <protocol>vnc</protocol>
  <param name="hostname">localhost</param>
  <param name="port">5900</param>
  <param name="password">VNCPASS</param>
</authorize>

<!-- Another user, but using md5 to hash the password
      (example below uses the md5 hash of "PASSWORD") -->
<authorize
  username="USERNAME2"
  password="319f4d26e3c536b5dd871bb2c52e3178"
  encoding="md5">
```

```
<!-- First authorized connection -->
<connection name="localhost">
  <protocol>ssh</protocol>
  <param name="hostname">localhost</param>
  <param name="port">22</param>
</connection>

      <!-- authorized connection -->
<connection name="ubuntu-vnc">
  <protocol>vnc</protocol>
  <param name="hostname">192.168.128.200</param>
  <param name="port">5901</param>
  <param name="password">123456</param>
</connection>

<connection name="ubuntu-ssh">
  <protocol>ssh</protocol>
  <param name="hostname">192.168.128.200</param>
  <param name="port">22</param>
</connection>
```

可以进行SSH、RDP、VNC、TELENT的远程桌面:

```
<connection name="windows-telnet">
  <protocol>telnet</protocol>
  <param name="hostname">192.168.128.201</param>
  <param name="port">23</param>
<!-- <param name="username">administrator</param> -->
<!-- <param name="password">123456</param> -->
</connection>

<connection name="windows-vnc">
  <protocol>vnc</protocol>
  <param name="hostname">192.168.128.201</param>
  <param name="port">5900</param>
  <param name="password">123456</param>
</connection>

<connection name="windows-rdp">
  <protocol>rdp</protocol>
  <param name="hostname">192.168.128.201</param>
  <param name="port">3389</param>
<!-- <param name="username">administrator</param> -->
<!-- <param name="password">123456</param> -->
<!-- <param name="domain">XP-201505312237</param> -->
</connection>
```

这就是登录界面了，用户名密码就是XML里配置的：

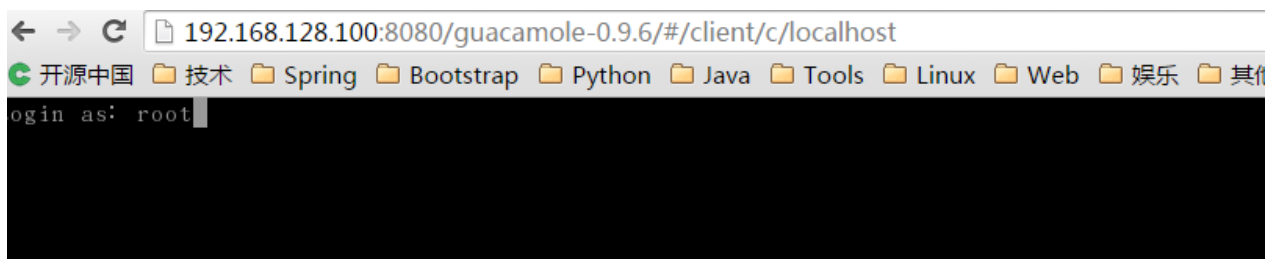
The image shows the Guacamole 0.9.6 login interface. At the top is the Guacamole logo, which is a stylized green and black circular icon. Below the logo, the text "GUACAMOLE 0.9.6" is displayed in a bold, black, sans-serif font. Underneath the text are two input fields: the first is labeled "USERNAME2" and the second contains a series of dots, indicating a password field. Both input fields have a light yellow background. At the bottom of the interface is a dark gray button with the word "Login" in white, bold, sans-serif font.

主界面已经把我在xml里配置的主机都列出来了。

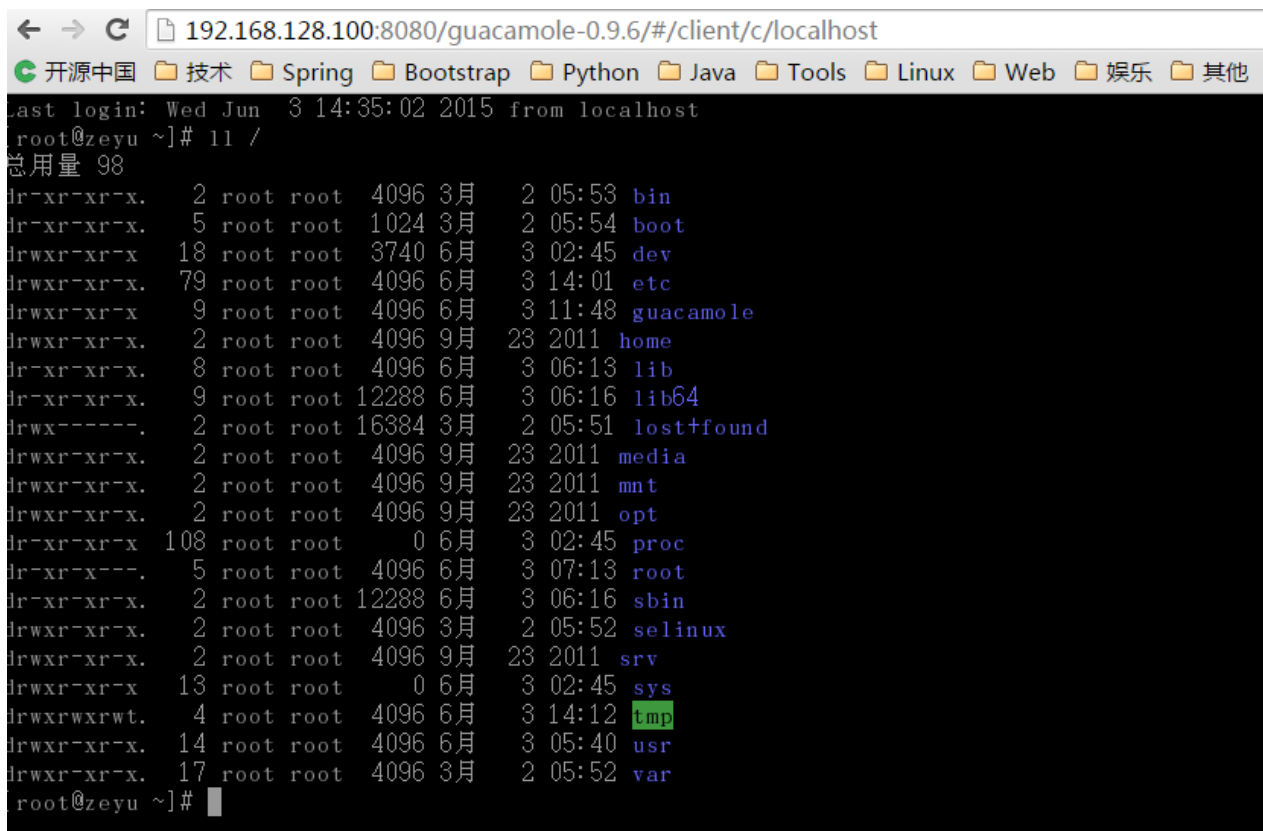


①. 本地主机 (Linux) SSH测试

只需配置host和port，其它和shell工具一样：



可以命令、可以Tab补全，最牛的是可以VI，基本可以替换shell工具了：



②. 远程主机 (Linux) SSH测试

这里用了一个远程的Ubuntu，也只需要主机和端口号：

```

192.168.128.100:8080/guacamole-0.9.6/#/client/c/ubuntu-ssh
开源中国 技术 Spring Bootstrap Python Java Tools Linux Web 娱乐 其他
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-35-generic x86_64)

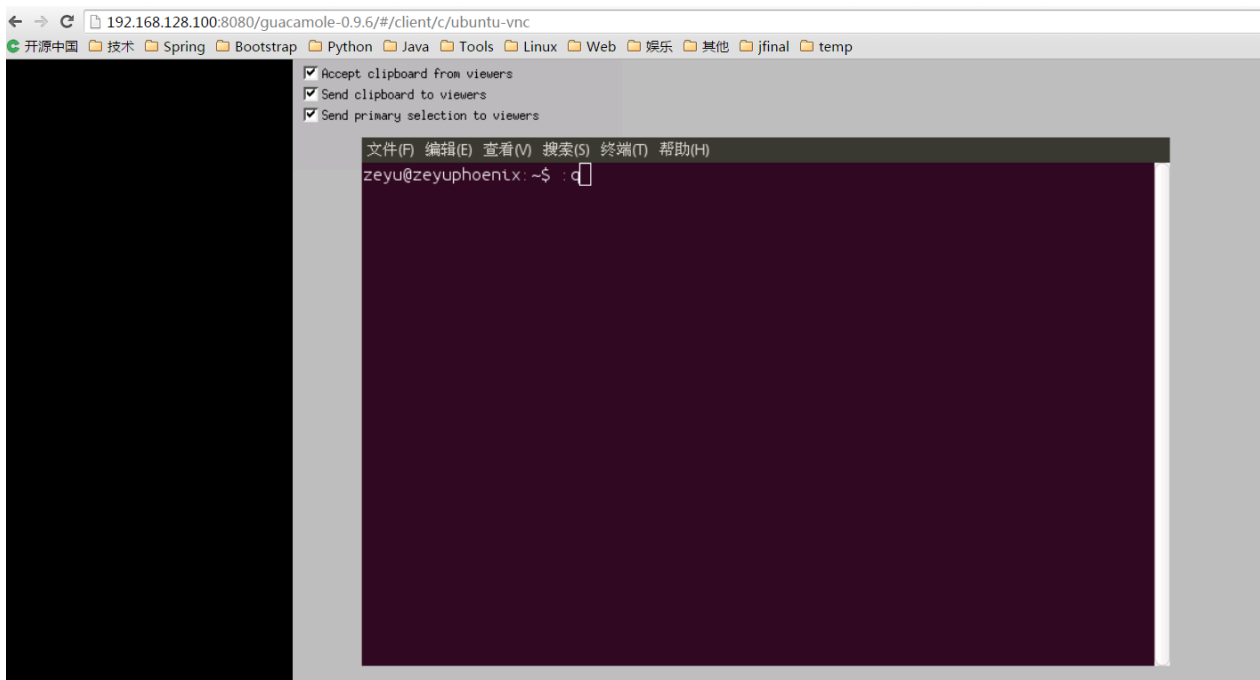
* Documentation:  https://help.ubuntu.com/

Last login: Tue Jun  2 23:07:05 2015 from 192.168.128.100
zeyu@zeyuphoenix:~$ ll
总用量 256
drwxr-xr-x 19 zeyu zeyu  4096  5月 31 17:02 ./
drwxr-xr-x  3 root root  4096  9月 12  2014 ../
-rw-r--r--  1 zeyu zeyu    72  9月 12  2014 .bash_history
-rw-r--r--  1 zeyu zeyu   220  9月 12  2014 .bash_logout
-rw-r--r--  1 zeyu zeyu  3637  9月 12  2014 .bashrc
drwx----- 17 zeyu zeyu  4096  5月 31 17:00 .cache/
drwx-----  3 zeyu zeyu  4096  9月 12  2014 .compiz/
drwx----- 14 zeyu zeyu  4096  9月 12  2014 .config/
drwx-----  3 zeyu zeyu  4096  5月 31 17:02 .dbus/
-rw-r--r--  1 zeyu zeyu    25  9月 12  2014 .dmrc
-rw-r--r--  1 zeyu zeyu  8980  9月 12  2014 examples.desktop
drwx-----  3 zeyu zeyu  4096  5月 31 17:02 .gconf/
-rw-rw-r--  1 zeyu zeyu 134276  9月 12  2014 hs_err_pid12647.log
-rw-r--r--  1 zeyu zeyu   1014  5月 31 16:45 .ICEauthority
drwxr-xr-x  3 zeyu zeyu  4096  9月 12  2014 .local/
drwxrwxr-x  3 zeyu zeyu  4096  9月 12  2014 .netbeans/
-rw-r--r--  1 zeyu zeyu    675  9月 12  2014 .profile
drwx-----  2 zeyu zeyu  4096  9月 12  2014 .sunpinyin/
drwxrwxr-x  2 zeyu zeyu  4096  5月 31 17:02 .vnc/
-rw-r--r--  1 zeyu zeyu   218  5月 31 17:02 .Xauthority
-rw-r--r--  1 zeyu zeyu   132  5月 31 16:45 .xsession-errors
-rw-r--r--  1 zeyu zeyu   132  9月 12  2014 .xsession-errors.old
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 公共的/
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 模板/
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 视频/
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 图片/
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 文档/
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 下载/
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 音乐/
drwxr-xr-x  2 zeyu zeyu  4096  9月 12  2014 桌面/

```

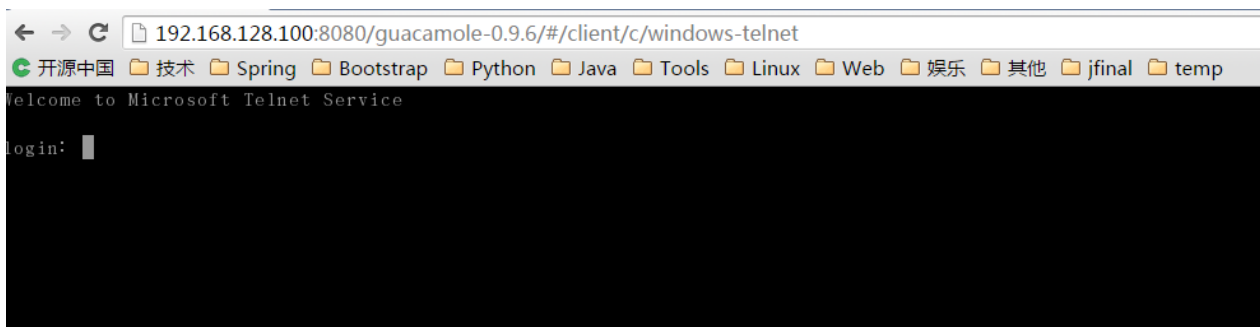
③. 远程主机 (Linux) VNC测试

这里我用的是ubuntu，安装了vnc4server，关于vnc server的安装就省略了：

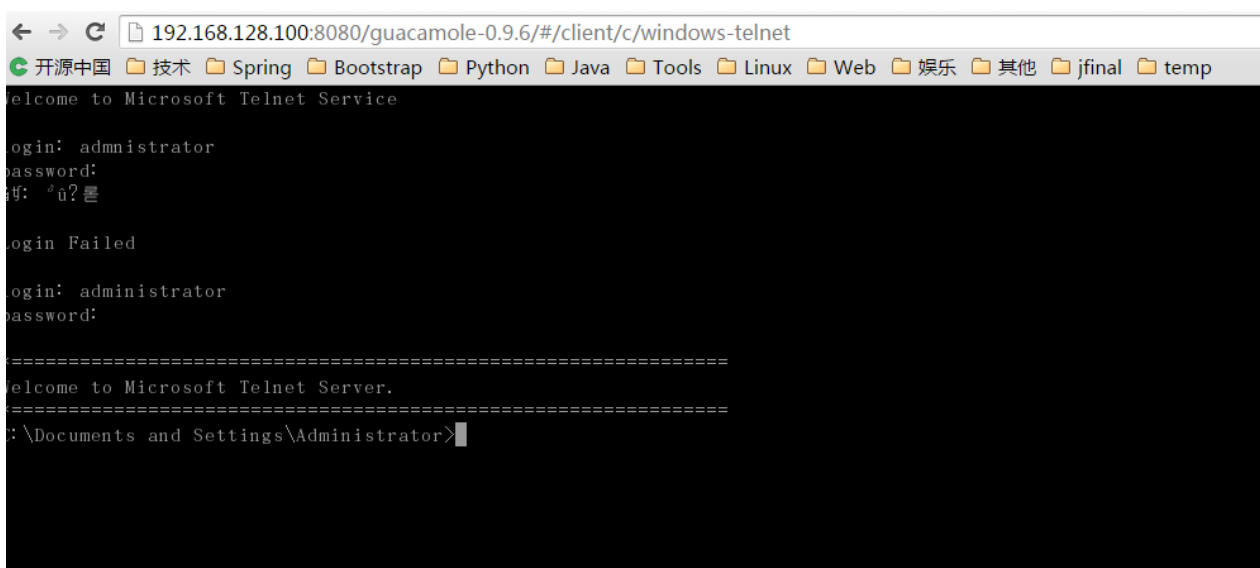


④. 远程主机 (Windows) TELNET测试

Telnet不安全，官方不推荐使用了，这里只是测试:



输入用户名和密码就进入了远程主机的dos窗口:



⑤. 远程主机 (Windows) RDP测试

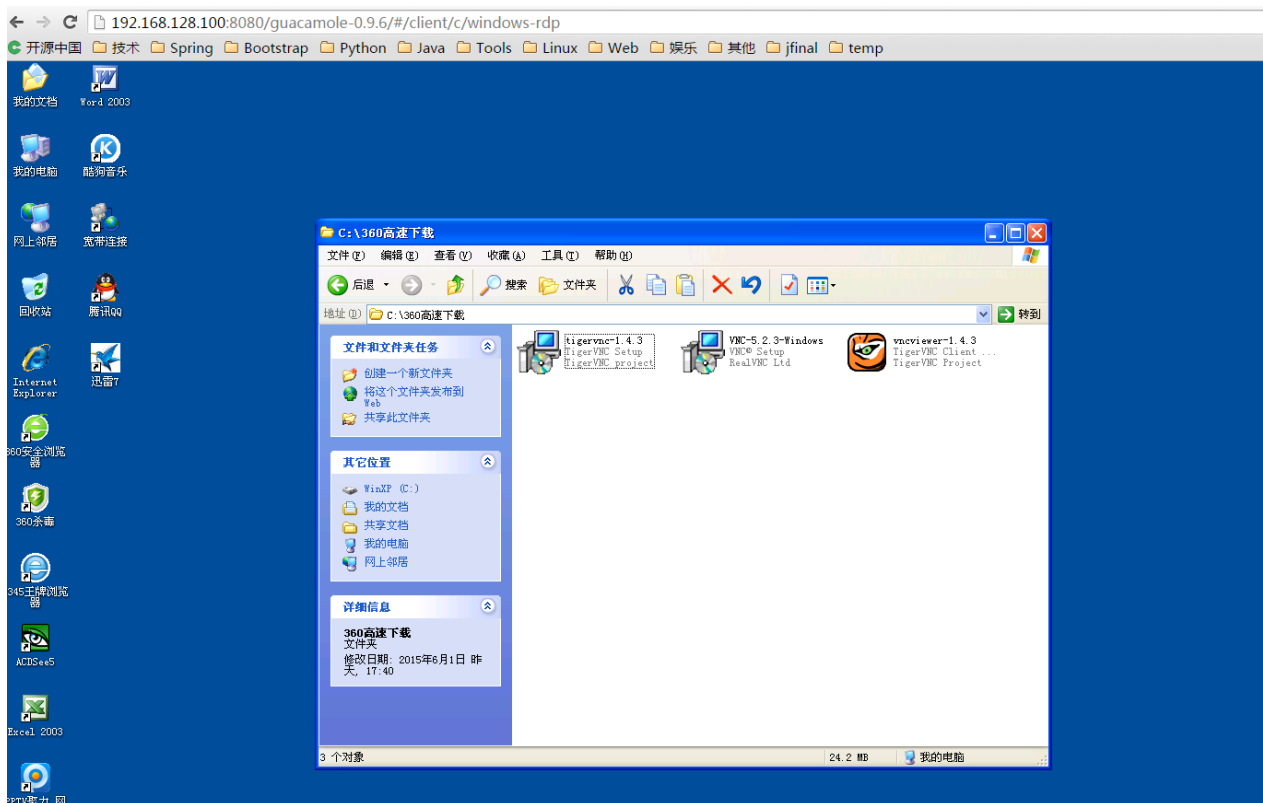
RDP可以只需要主机和端口号，会出现输入用户和密码的界面，如果配置了用户和密码，会直接进入系统：



输入我们的用户名和密码就可以了：



远程WEB桌面，操作还很流畅的。

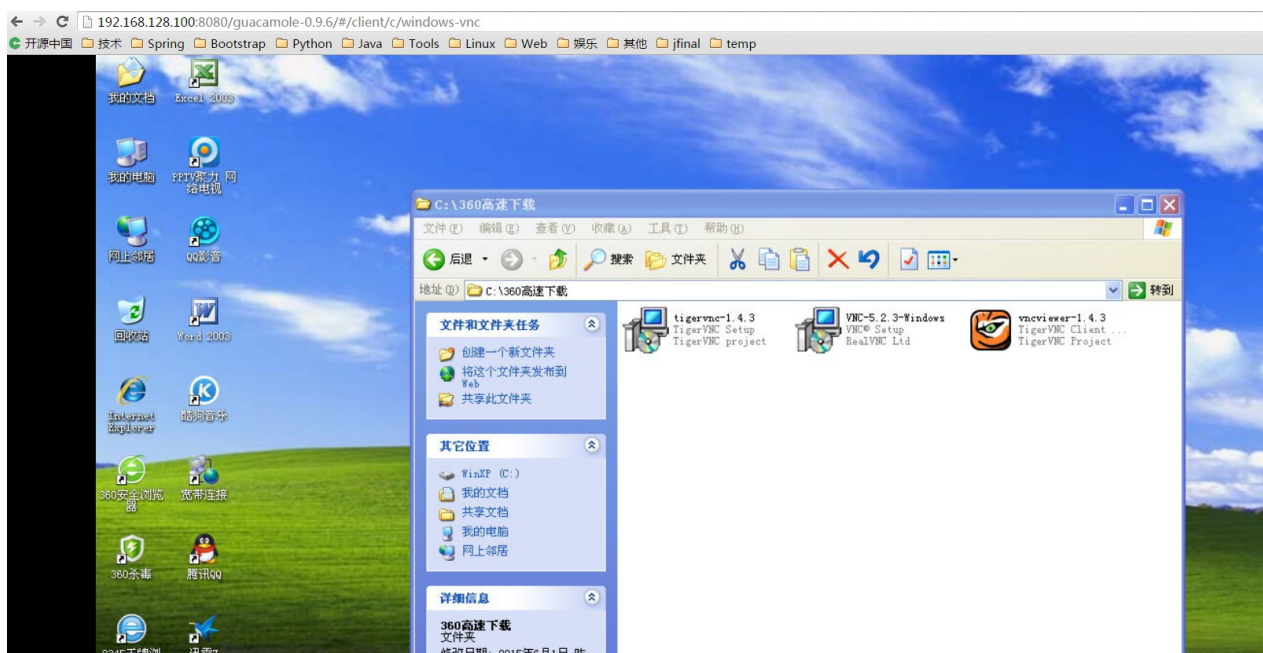


⑥. 远程主机 (Windows) VNC测试

Windows因为有了远程桌面，所以VNC只是测试，但是有很多坑啊。

我先用了RealVNC，直接下了专业版，我去，只有它自己的客户端可以连。

后来又用了 TigerVNC连接，页面很卡，后来网上有人说需要把JPEG关掉，页面好点了，但是没有RDP好，而且有注销问题



2. NoAuth模式测试

NoAuth只是测试，因为没有密码，非常不安全：

配置文件如下所示:

```
[root@zeyu guacamole]# vi noauth/guacamole.properties
```

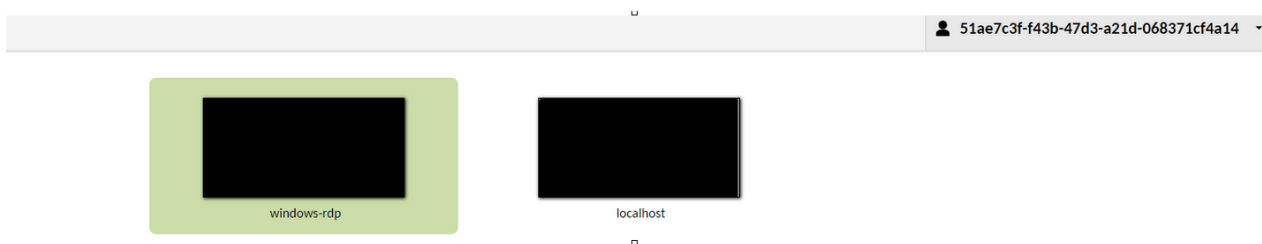
```
# Hostname and port of guacamole proxy
guacd-hostname: localhost
guacd-port: 4822

# Auth provider class (authenticates user/pass combination, needed if using the provided login screen)
auth-provider: net.sourceforge.guacamole.net.auth.noauth.NoAuthenticationProvider
# NoAuth properties
noauth-config: /etc/guacamole/noauth-config.xml
```

配置基本和XML一样，只是tag标签不同:

```
<configs>
  <config name="localhost" protocol="ssh">
    <param name="hostname" value="localhost" />
    <param name="port" value="22" />
  </config>
  <config name="windows-vnc" protocol="vnc">
    <param name="hostname" value="192.168.128.201" />
    <param name="port" value="5900" />
  </config>
  <config name="windows-rdp" protocol="rdp">
    <param name="hostname" value="192.168.128.201" />
    <param name="port" value="3389" />
  </config>
</configs>
```

只是变了随机的用户名，其它都是一样的



3. JDBC-Mysql模式测试

JDBC模式需要配置host、port、username和password，需要把sql导入数据库:

```
[root@zeyu guacamole]# vi mysql/guacamole.properties
```

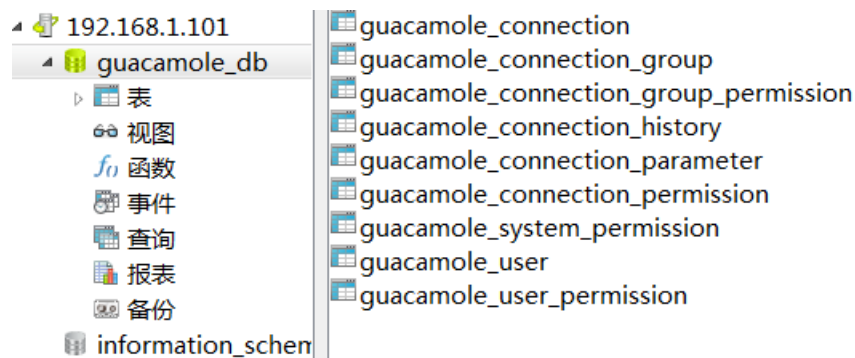


```
# Hostname and port of guacamole proxy
guacd-hostname: localhost
guacd-port: 4822

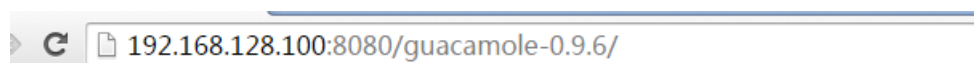
# Auth provider class
auth-provider: net.sourceforge.guacamole.net.auth.mysql.MySQLAuthenticationProvider

# MySQL properties
mysql-hostname: 192.168.1.101
mysql-port: 3306
mysql-database: guacamole_db
mysql-username: root
mysql-password: root
```

数据库表如下所示:





登录界面就是user表里的用户名和密码:



随手配置了2个:


ALL CONNECTIONS

 192.168.128.201
 localhost-ssh

配置也有界面了，不用修改xml文件了，直接根据远程主机配置就可以了，最后保存数据库:

Name:
Location:
Protocol:

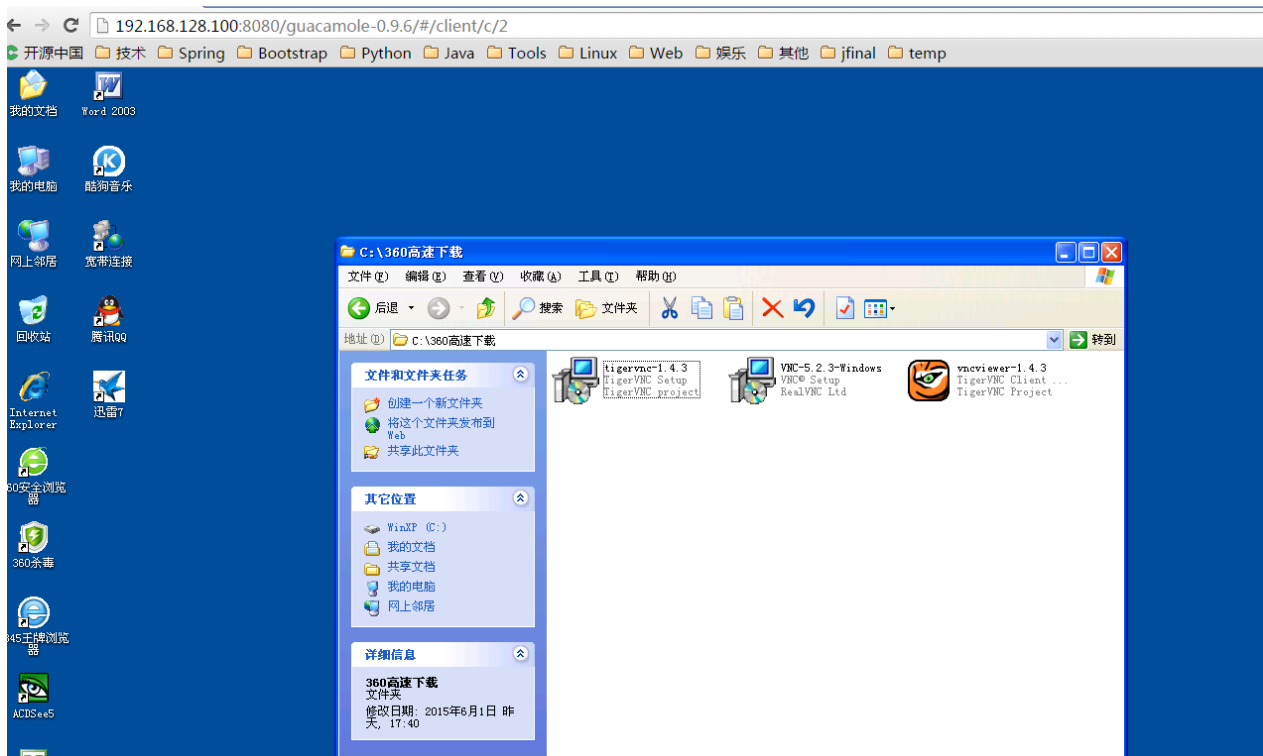
PARAMETERS

Hostname:
Port:
Password: 
Read-only: ☐
Swap red/blue components: ☐
Cursor:
Color depth:
Repeater destination host:
Repeater destination port:
Enable audio: ☐
Audio server name:

连接效果一样:

```
Last login: Wed Jun  3 07:49:11 2015 from localhost
[root@zeyu ~]# 11
总用量 24
-rw-----. 1 root root 1253 3月  2 05:55 anaconda-ks.cfg
-rw-r--r--. 1 root root 13231 3月  2 05:55 install.log
-rw-r--r--. 1 root root 3482 3月  2 05:54 install.log.syslog
[root@zeyu ~]# cd /
[root@zeyu /]# 11
总用量 98
dr-xr-xr-x.  2 root root  4096 3月  2 05:53 bin
dr-xr-xr-x.  5 root root  1024 3月  2 05:54 boot
drwxr-xr-x 18 root root  3740 6月  3 02:45 dev
drwxr-xr-x 79 root root  4096 6月  3 14:01 etc
drwxr-xr-x  9 root root  4096 6月  3 11:48 guacamole
drwxr-xr-x.  2 root root  4096 9月 23 2011 home
dr-xr-xr-x.  8 root root  4096 6月  3 06:13 lib
dr-xr-xr-x.  9 root root 12288 6月  3 06:16 lib64
drwx-----. 2 root root 16384 3月  2 05:51 lost+found
drwxr-xr-x.  2 root root  4096 9月 23 2011 media
drwxr-xr-x.  2 root root  4096 9月 23 2011 mnt
drwxr-xr-x.  2 root root  4096 9月 23 2011 opt
dr-xr-xr-x 110 root root    0 6月  3 02:45 proc
dr-xr-xr-x.  5 root root  4096 6月  3 07:13 root
dr-xr-xr-x.  2 root root 12288 6月  3 06:16 sbin
drwxr-xr-x.  2 root root  4096 3月  2 05:52 selinux
drwxr-xr-x.  2 root root  4096 9月 23 2011 srv
drwxr-xr-x 13 root root    0 6月  3 02:45 sys
drwxrwxrwt.  4 root root  4096 6月  3 14:12 tmp
drwxr-xr-x. 14 root root  4096 6月  3 05:40 usr
drwxr-xr-x. 17 root root  4096 3月  2 05:52 var
[root@zeyu /]# █
```

VNC-Windows也测试一下:



五. 其它工具

其它远程虚拟桌面基于web的还有noVNC，而OpenStack 就是用noVNC。

noVNC的工作原理：noVNC提供一种在网页上通过html5的Canvas，访问机器上vncserver提供的vnc服务，需要做tcp到 websocket的转化，才能在html5中显示出来。网页就是一个客户端，类似win或linux下面的vncviewer，只是此时填的不是裸露的vnc服务的ip+port，而是由noVNC提供的websockets的代理，在noVNC代理服务器上要配置每个vnc服务，noVNC提供一个标识，去反向代理所配置的vnc服务。

noVNC主要问题是目前不支持其它的协议，比如RDP、NX以及Red Hat的Spice协议