Preparación Máquina Virtual

Para ejecutar la herramienta de Benchmarking es necesario acoplar una máquina virtual que se adecue al funcionamiento de la aplicación. El siguiente documento explica cuáles son los principales elementos a configurar en la máquina virtual a configurar.

# Sistema Operativo

El desarrollo y ejecución exitoso se probó en un sistema operativo Debian 8.5.0, usted puede encontrar los archivos de configuración en el siguiente hipervínculo: <https://www.debian.org/CD/http-ftp/>

Sin embargo, cualquier sistema operativo Linux, Red Hat Linux y Debian GNU/Linux tiene la compatibilidad de librerías y herramientas para la configuración de la máquina virtual cliente. Para la instalación siga las instrucciones del archivo de instalación utilizado (puede ver un ejemplo en el anexo “Capturas de Pantalla en la instalación de Sistema Debian 8.5”).

# Configuración de Permisos

Se recomienda hacer uso del paquete de configuración *sudo*, si su sistema operativo no lo incluye, instalando con el siguiente comando.

aptitude install sudo

Además, es necesario configurar sudo para el usuario en dónde se ejecutará la aplicación cliente.

En el siguiente hipervínculo, se encuentran las instrucciones para realizar la configuración en debian.

<https://www.privateinternetaccess.com/forum/discussion/18063/debian-8-1-0-jessie-sudo-fix-not-installed-by-default>

# Instalación de Guest Additions

Para la instalación de las Guest Additions en Debian y Ubuntu:

En la máquina virtual, asegurarse de que no haya un CDRom en la unidad de disco, y desde la línea de comandos, ejecutar lo siguiente como superusuario:

apt-get update

apt-get upgrade

apt-get install build-essential

apt-get install linux-headers-$(uname -r) dkms

apt-get install module-assistant

apt-get install xserver-xorg xserver-xorg-core

m-a prepare

mount /media/cdrom

cd /media/cdrom

sh VBoxLinuxAdditions.run

Algunas explicaciones se pueden encontrar en las siguientes fuentes:

* <http://geekland.eu/instalar-las-guest-additions-debian-derivadas/>
* <https://diversidadyunpocodetodo.blogspot.com.co/2013/07/virtualbox-guest-additions-debian-wheezy-distribuciones-derivadas-point-linux.html>

# Configuración de GRUB

Se puede ajustar el tiempo de espera del manejador de ejecución del sistema operativo GRUB. Esto con el fin de disminuir el tiempo de carga de la máquina virtual.

Ejecutar la siguiente instrucción

sudo gedit /etc/default/grub

Y escribir la contraseña del usuario.

Se mostrará un archivo con los siguientes elementos:

GRUB\_DEFAULT=0

GRUB\_HIDDEN\_TIMEOUT=0

GRUB\_HIDDEN\_TIMEOUT\_QUIET=true

GRUB\_TIMEOUT=1

GRUB\_DISTRIBUTOR=`lsb\_release -i -s 2> /dev/null || echo Debian`

GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet splash"

GRUB\_CMDLINE\_LINUX=""

Asegúrese que los parámetros *GRUB\_DEFAULT*, *GRUB\_HIDDEN\_TIMEOUT* sean iguales a 0 y *GRUB\_TIMEOUT* igual a 1. Al salir oprima CTRL + S y CTRL + Q.

Más información:

<http://askubuntu.com/questions/148095/how-do-i-set-the-grub-timeout-and-the-grub-default-boot-entry>

# Instalación de JAVA

Se requiere instalar Java para la ejecución de la aplicación cliente. Para tal propósito puede seguir las siguientes instrucciones:

1. Conozca el tipo de sistema operativo ejecutado (32 bits/64 bits).

uname -m

Si x86\_64 entonces es kernel de 64 bits. Si i686 entonces es kernel de 32 bits.

1. Conociendo el tipo de sistema operativo, descargue el instalable y la versión de Java, como recomendación instale la versión Java 8.5

La siguiente instrucción instalará la versión Java JDK 8.5 para un sistema de 64 bits. Asegúrese de cambiar la URL si desea utilizar otra versión.

wget --header "Cookie: oraclelicense=accept-securebackup-cookie" http://download.oracle.com/otn-pub/java/jdk/8u5-b13/jdk-8u5-linux-x64.tar.gz

1. Ahora para la instalación del paquete descargado, ejecute el siguiente código

sudo su

mkdir /opt/jdk

tar -zxf jdk-8u5-linux-x64.tar.gz -C /opt/jdk

ls /opt/jdk

Se crea un directorio para el jdk, y se extrae el archivo en la carpeta, al final se consulta si la extracción ha sido exitosa. Recuerde cambiar el nombre del archivo y la ubicación si hizo algún cambio en el proceso.

1. Ahora se ajusta la versión descarga como la versión por defecto.

update-alternatives --display java

update-alternatives --display javac

Si desea conocer más acerca de la instalación, puede consultar la siguiente fuente:

<https://www.digitalocean.com/community/tutorials/how-to-manually-install-oracle-java-on-a-debian-or-ubuntu-vps>

# Instalación de Stress NG

Para realizar la carga al sistema se hace uso de la herramienta Stress NG. Instale de la siguiente forma:

$ sudo apt-get install stress-ng [on Debian based systems]

# yum install stress-ng [on RedHat based systems]

Otras fuentes de Instalación de Stress NG en Linux son las siguientes:

* <http://www.tecmint.com/linux-cpu-load-stress-test-with-stress-ng-tool/2/>
* <http://www.cyberciti.biz/faq/stress-test-linux-unix-server-with-stress-ng/>

Si desea conocer sobre la herramienta, consulte esta fuente:

* <http://kernel.ubuntu.com/~cking/stress-ng/>

# Configuración de JAR cliente

Finalmente, exporte el JAR del cliente al directorio de su elección.

Además, es necesario configurar el archivo ubicado en */etc/rc.local*

sudo nano /etc/rc.local

Escriba la contraseña del usuario.

Dos líneas antes del final escriba la siguiente línea

java –jar /path/client.jar 192.168.0.XXX 20000

Dónde:

* path es el directorio dónde está ubicado el jar.
* Client el nombre que tiene el jar.
* 192.168.0.XXX la dirección IP del servidor, asegúrese de la apertura de redes en la configuración de la máquina virtual.
* 20000 el puerto elegido para comunicación

Guardar al salir usando los comandos CTRL + S y salir con CTRL + Q

# Scripts de Carga

Ahora, en la carpeta dónde está ubicado el jar del cliente es necesario crear los siguientes scripts

ALL

**all.sh**

#!/bin/bash

rm stress-\*

stress-ng –all 1 &

MEMORY

**memory.sh**

#!/bin/bash

MEMORY=$(awk ‘/MemFree/{printf “%d\n”, $2}’ < /proc/meminfo)k

Rm stress-\*

stress-ng –vm 1 –vm-bytes $MEMORY &

CPU

**cpu.sh**

#!/bin/bash

CORES=&(grep ‘cpu cores’ /proc/cpuinfo | head –n1 | cut –f 2 –d’:’)

LOAD=100

rm stress-\*

stress-ng –cpu $CORES –cpu-load $LOAD &

DISK

**disk.sh**

#!/bin/bash

rm stress-\*

stress-ng –hdd 1 &

IO

**io.sh**

#!/bin/bash

rm stress-\*

stress-ng –-io 4 &

Los scripts pueden ser configurados al ajuste del usuario, si requiere más o menos carga, se puede editar siguiendo la librería de Stress NG.

# Anexos

## Capturas de Pantalla en la Instalación de sistema Debian 8.5

Hostname: debian

Root Password: debianUnaCloud

Username: user

Password User: userUnaCloud























































