

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sistemas operativos 1

Proyecto

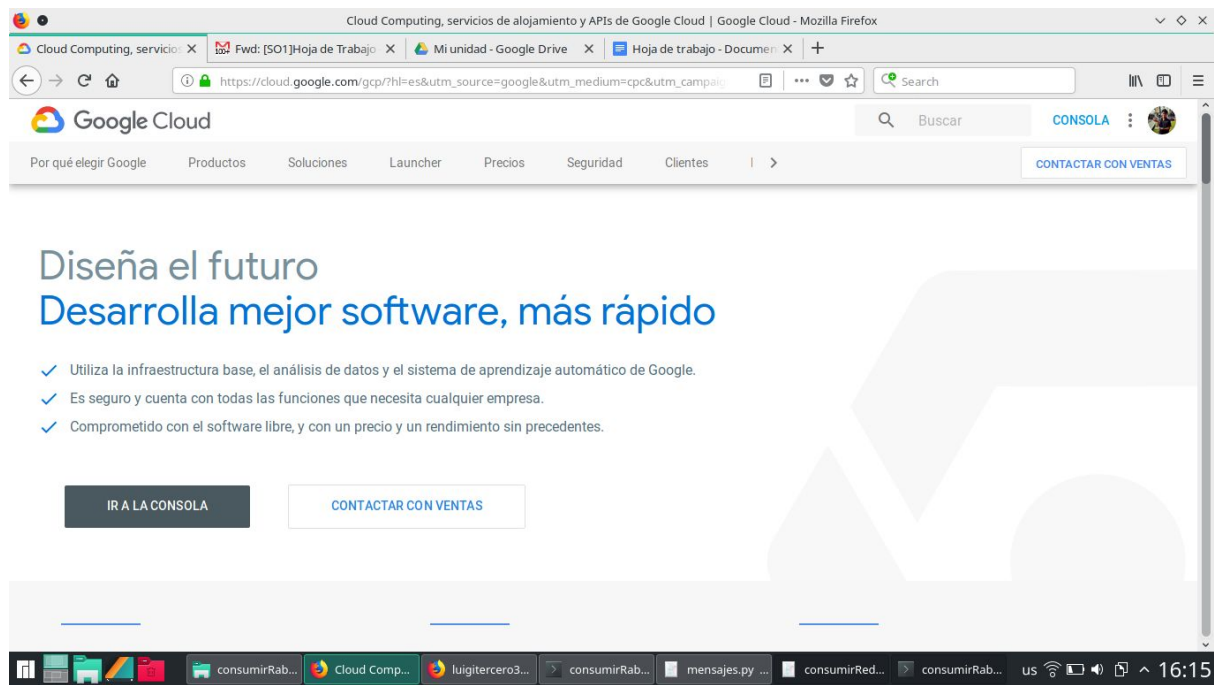
MANUAL TÉCNICO

Kevin Alberto Morán Orellana	-- 201404412
Luis Estuardo Azurdía Cárcamo	-- 201408606

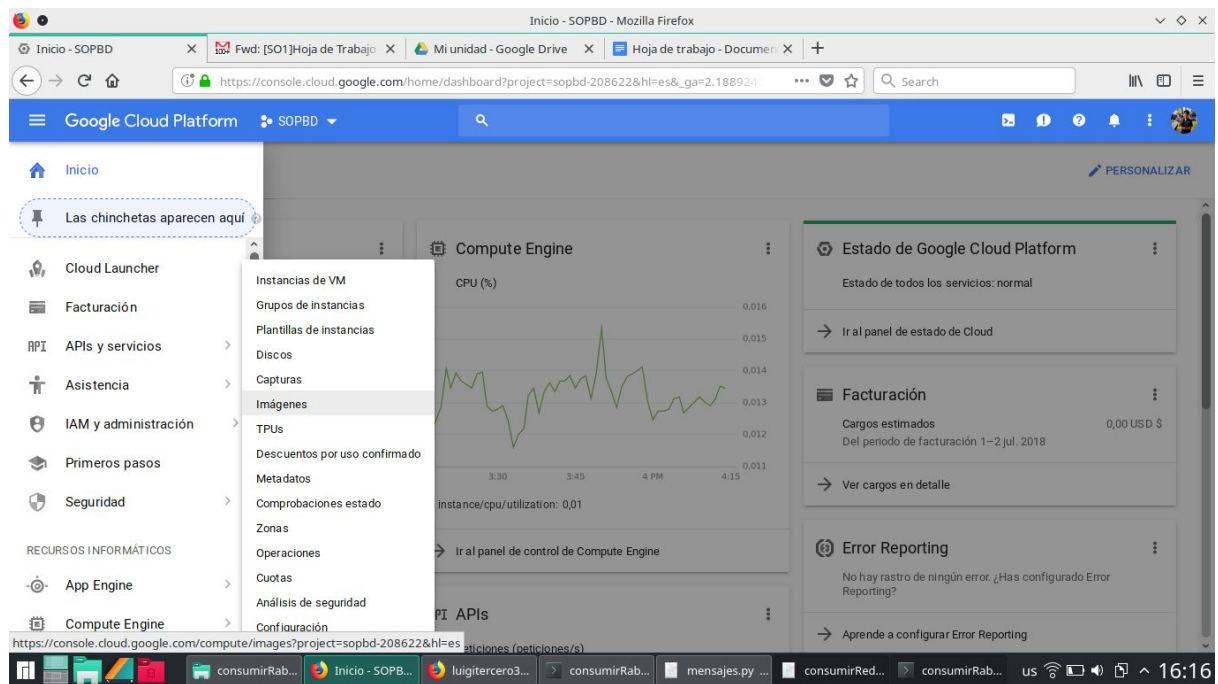
Crear una máquina virtual desde google cloud	2
Instalación de apache	7
Instalar apache TOMCAT	7
Instalación Payara	10
SSH Local	12
Instalar haproxy	13
Se agrega al repositorio las nueva librerías	13
Actualizar los repositorios	13
Instalar haproxy	14
Consultar versión	14
Crear como un back up de los archivos por defecto	14
En bin agregar el balanceador de carga	14
Reiniciar el servidor	14
Consultar página de balanceo	14
Agregar password	15
Variable de sesión y cache	15
Instalación Cassandra en Ubuntu 14.04	16
Actualizar base de datos	16
Instalar JDK 8	16
Install Apache Cassandra	18
APACHE BENCHMARK windows 10	19
Intalacion de xamp :	
http://www.mclibre.org/consultar/php/otros/xampp-instalacion-windows.html	19

Crear una máquina virtual desde google cloud

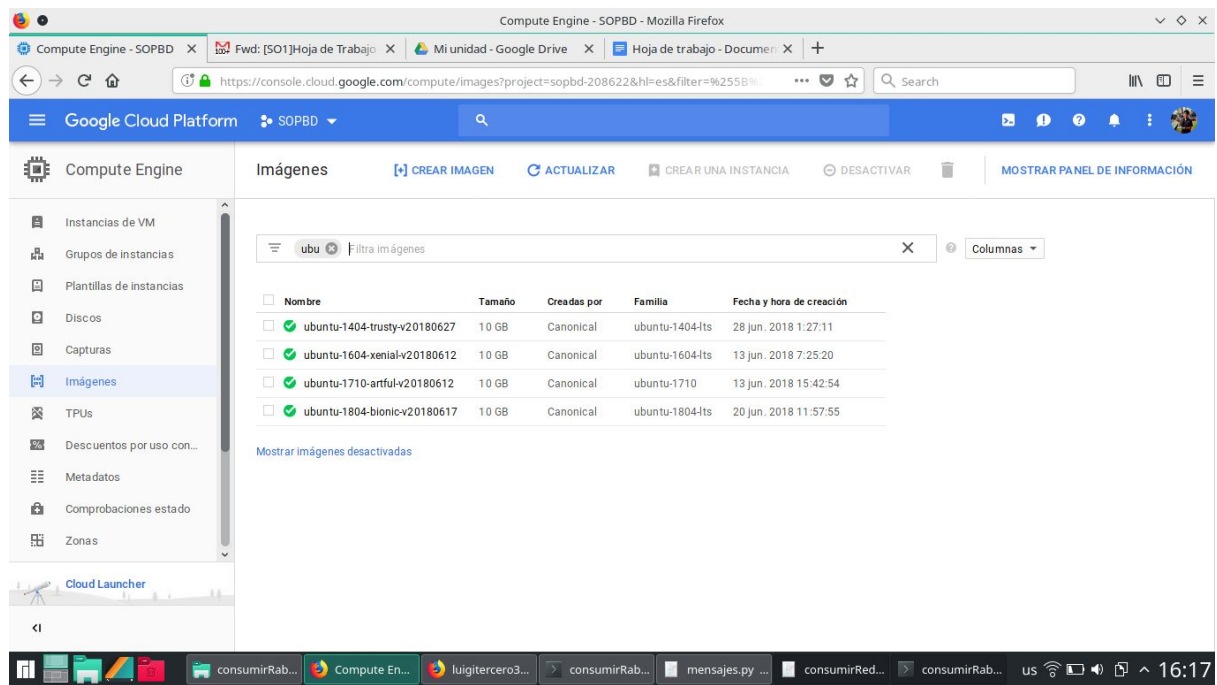
Dirigirse a consola



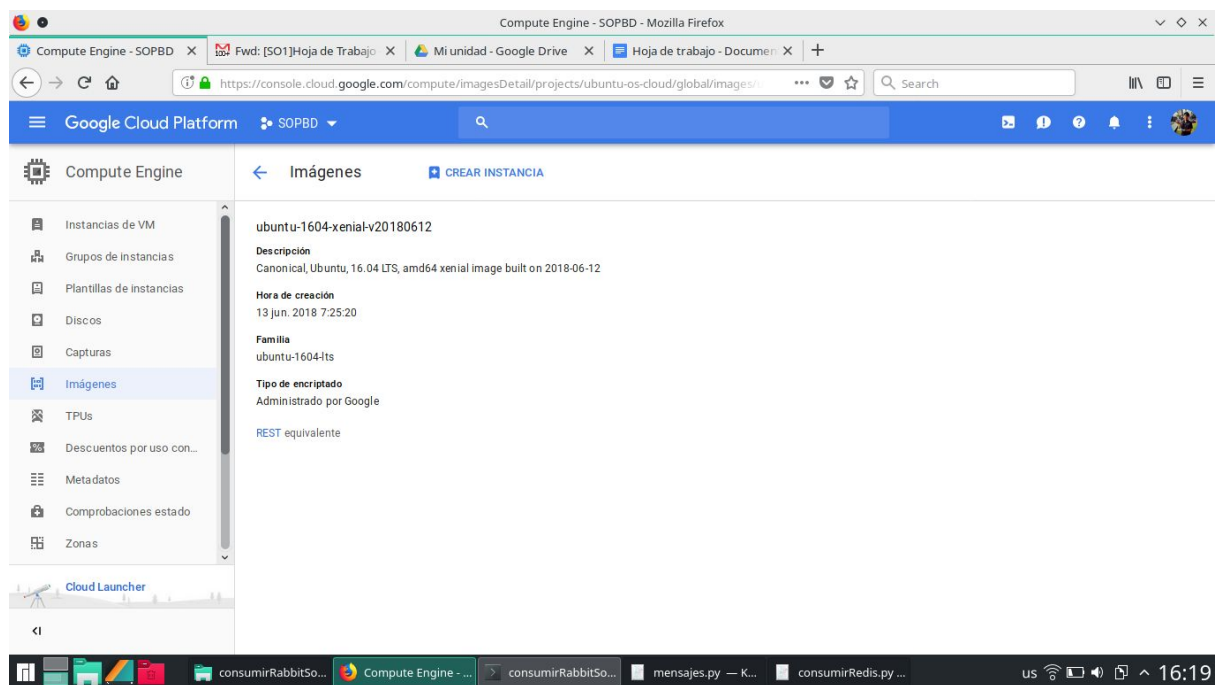
En menú de navegación dirigirse a comput engine-> imágenes



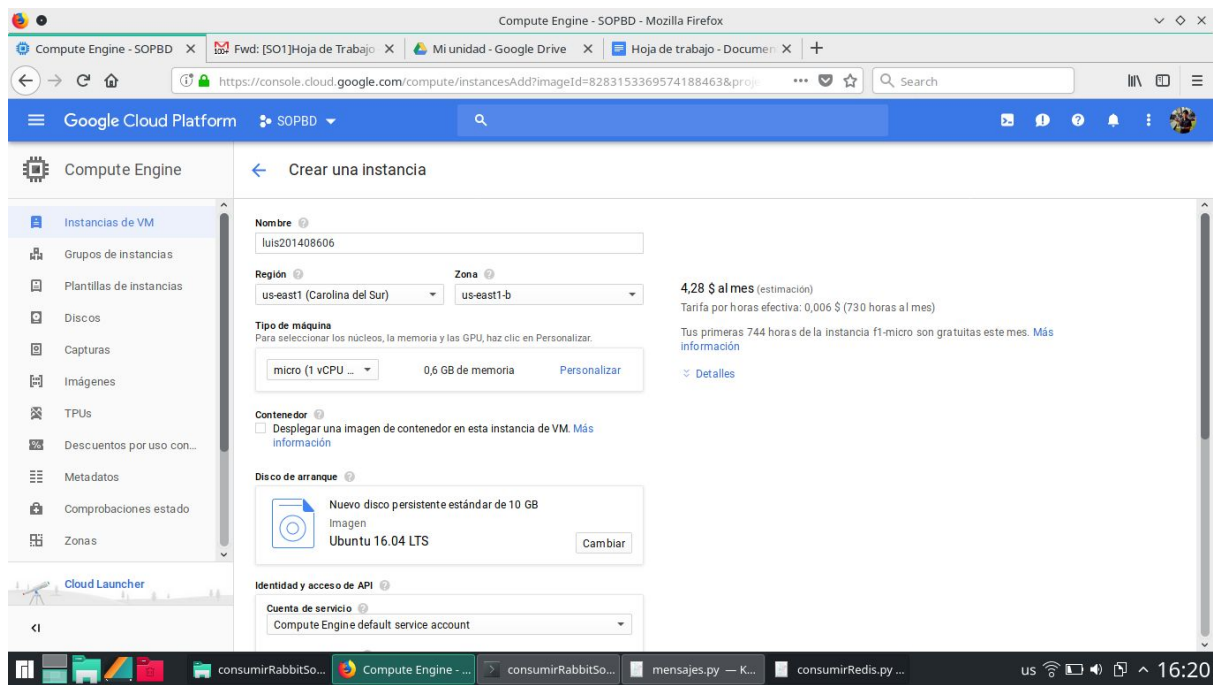
Filtrar los datos



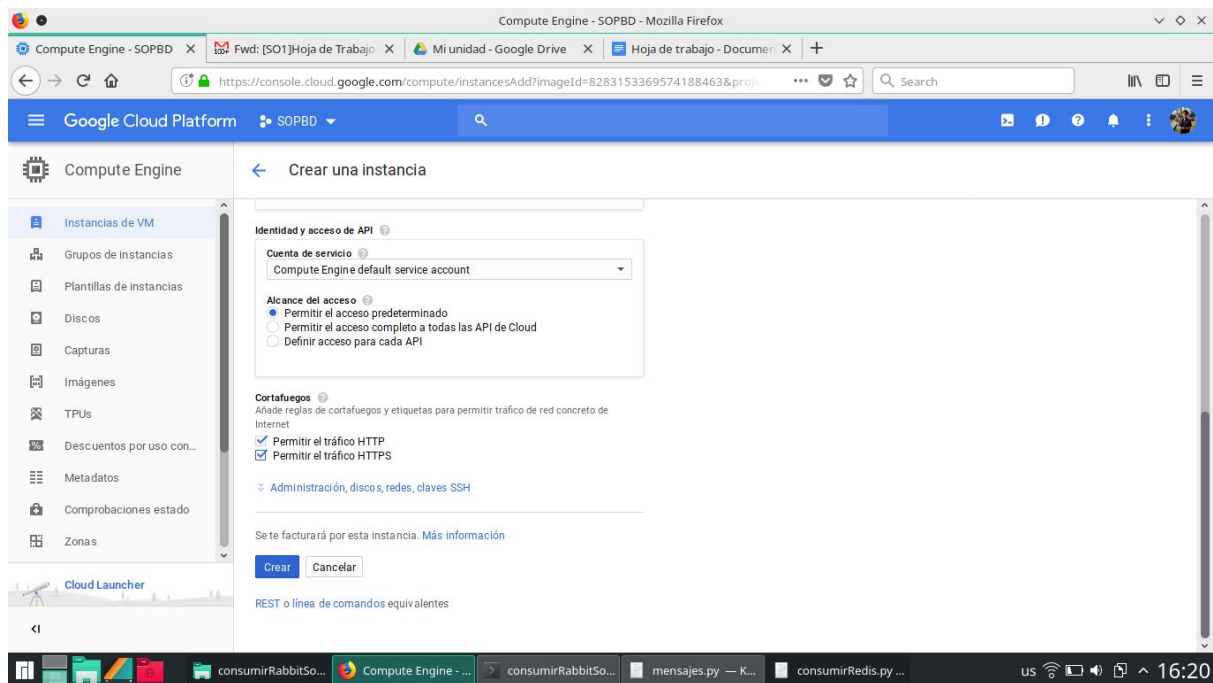
Seleccionar la instancia
Ver instancia



Crear el nombre de la instancia



Crea la instancia



Compute Engine - SOPBD - Mozilla Firefox

Compute Engine - SOPBD x Fwd: [SO1]Hoja de Trabajo x Mi unidad - Google Drive x Manjaro Linux x +

https://console.cloud.google.com/compute/instances?project=sopbd-208622&hl=es

Google Cloud Platform SOPBD

Compute Engine

Instancias de VM

CREAR INSTANCIA IMPORTAR VM MOSTRAR PANEL DE INFORMACIÓN

La instancia 'colitas' está infrutilizada. Ahorrarás aproximadamente 16 \$ al mes si cambias al tipo de máquina g1-small (1 vCPU, 1,7 GB de memoria). Más información Ignorar

Filtrar las instancias de VM

Nombre	Zona	Recomendación	IP interna	IP externa	Conectar
colitas	us-east1-b	Ahorra 16 \$ al mes	10.142.0.3 (nic0)	35.229.58.120	SSH
haproxy	us-east1-b		10.142.0.4 (nic0)	35.229.32.168	SSH
luis201408606	us-east1-b			Ninguna	

Cloud Launcher

consumirRabbitSo... Compute Engine - ... consumirRabbitSo... mensajes.py - K... consumirRedis.py ...

us 16:22

Se puede entrar desde la consola de google cloud

Compute Engine - SOPBD - Mozilla Firefox

Compute Engine - SOPBD x Fwd: [SO1]Hoja de Trabajo x Mi unidad - Google Drive x Manjaro Linux x +

https://console.cloud.google.com/compute/instances?project=sopbd-208622&hl=es

Google Cloud Platform SOPBD

Compute Engine

Instancias de VM

CREAR INSTANCIA IMPORTAR VM MOSTRAR PANEL DE INFORMACIÓN

La instancia 'colitas' está infrutilizada. Ahorrarás aproximadamente 16 \$ al mes si cambias al tipo de máquina g1-small (1 vCPU, 1,7 GB de memoria). Más información Ignorar

Filtrar las instancias de VM

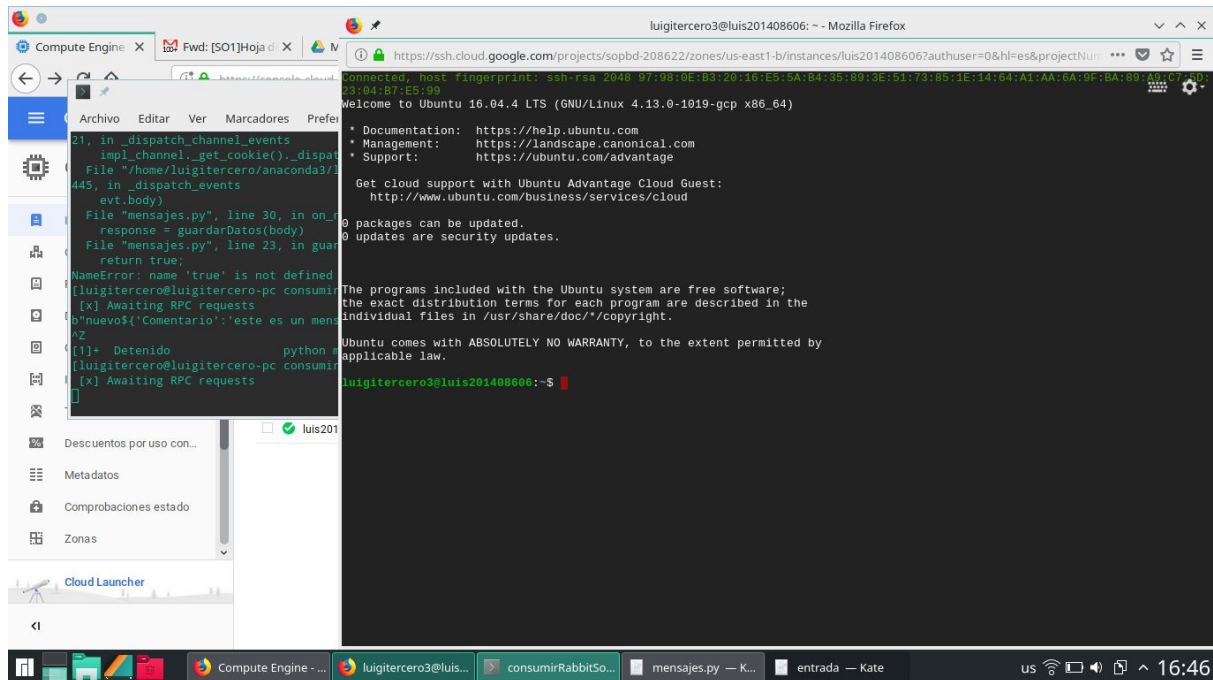
Nombre	Zona	Recomendación	IP interna	IP externa	Conectar
colitas	us-east1-b	Ahorra 16 \$ al mes	10.142.0.3 (nic0)	35.229.58.120	SSH
haproxy	us-east1-b		10.142.0.4 (nic0)	35.229.32.168	SSH
luis201408606	us-east1-b			Ninguna	

Cloud Launcher

consumirRabbitSo... Compute Engine - ... consumirRabbitSo... mensajes.py - K... consumirRedis.py ...

us 16:22

Instalación de apache



Instalar apache TOMCAT

```
sudo apt-get update  
sudo apt-get upgrade
```

instalar java:

```
sudo apt-get install default-jdk
```

Actualiza la JAVA_HOME variable

Encuentra el directorio para JAVA_HOME.

```
update-alternatives --config java
```

Copie el directorio y luego escriba lo siguiente.

```
nano /etc/environment
```

Esto abrirá el archivo que contiene variables de entorno.

Añadir JAVA_HOME.

```
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java"
```

La ruta entre comillas es la ruta que estás usando.

Cargar las variables de entorno

```
source /etc/environment
```

Verifique para asegurarse de que funcionó:

```
echo $JAVA_HOME
```

Crear grupo y usuario

```
groupadd tomcat
```

```
useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

Asegúrese de conocer la versión de Tomcat que desea instalar.

Crear directorio

```
mkdir /opt/tomcat
```

Descarga Tomcat

```
wget
```

```
http://www-eu.apache.org/dist/tomcat/tomcat-8/v8.5.32/bin/apache-tomcat-8.5.32.tar.gz
```

Descomprime los archivos en la carpeta tomcat.

```
tar xvf apache-tomcat-8*.tar.gz -C /opt/tomcat --strip-components=1
```

Asegúrese de que las carpetas tengan los permisos correctos.

```
cd /opt/
```

```
sudo chown -R tomcat tomcat/
```

Visualice la JAVA_HOME ruta nuevamente para que pueda copiarla.

```
nano /etc/systemd/system/tomcat.service
```

Configurar Tomcat

Copie y pegue la configuración siguiente en el tomcat.service archivo que acaba de crear. Edite la JAVA_HOME ruta a la que copió. Después de esto, guarde y cierre el archivo.

```
[Unit]
```

```
Description=Apache Tomcat Web Application Container
```

```
After=network.target
```

```
[Service]
```

```
Type=forking
```

```
Environment=JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre
```

```
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
```

```
Environment=CATALINA_HOME=/opt/tomcat
```



```
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server
-XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom'
```

```
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
```

```
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
```

```
[Install]
WantedBy=multi-user.target
```

Vuelva a cargar el daemon SystemD para que se incluyan las configuraciones de servicio que se agregaron.

```
sudo systemctl daemon-reload
```

Comience el servicio de Tomcat.

```
sudo systemctl start tomcat
```

Verifique el estado de su servidor Tomcat.

```
sudo systemctl status tomcat
```

MODIFICAR LOS SIGUIENTES ARCHIVOS PARA PERMITIR EL ACCESO A LAS APLICACIONES DE ADMINISTRACIÓN

```
sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
```

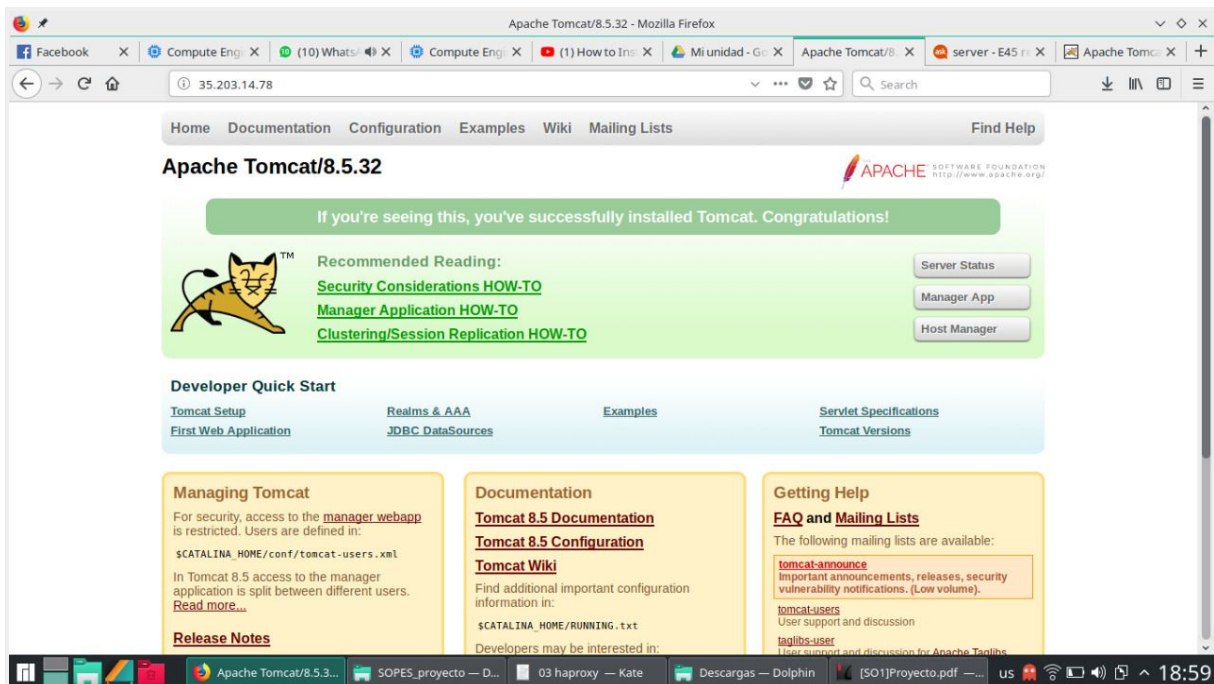
-----COMENTAR LA LÍNEA CON LA IP-----

```
<Context antiResourceLocking="false" privileged="true" >
  <!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />-->
</Context>
```

agregar usuario para poder ingresar al manger-gui

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

```
<role rolename="admin-gui"/>
<user username="admin" password="12345"
roles="manager-gui,admin-gui"/><role rolename="manager-gui"/>
```



Instalación Payara

Actualizar

```
sudo apt-get update  
sudo apt-get upgrade
```

Instalar java jdk

```
sudo apt-get install default-jdk
```

Actualiza la JAVA_HOME variable

Encuentra el directorio para JAVA_HOME.

```
update-alternatives --config java
```

Copie el directorio y luego escriba lo siguiente.

```
nano /etc/environment
```

Esto abrirá el archivo que contiene variables de entorno. Añadir JAVA_HOME.

```
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java"
```

La ruta entre comillas es la ruta que estás usando.

Cargar las variables de entorno

```
source /etc/environment
```

Verifique para asegurarse de que funcionó:

```
echo $JAVA_HOME
```

Crear directorio

```
mkdir /opt/payara  
cd /opt/payara
```

Descargar payara

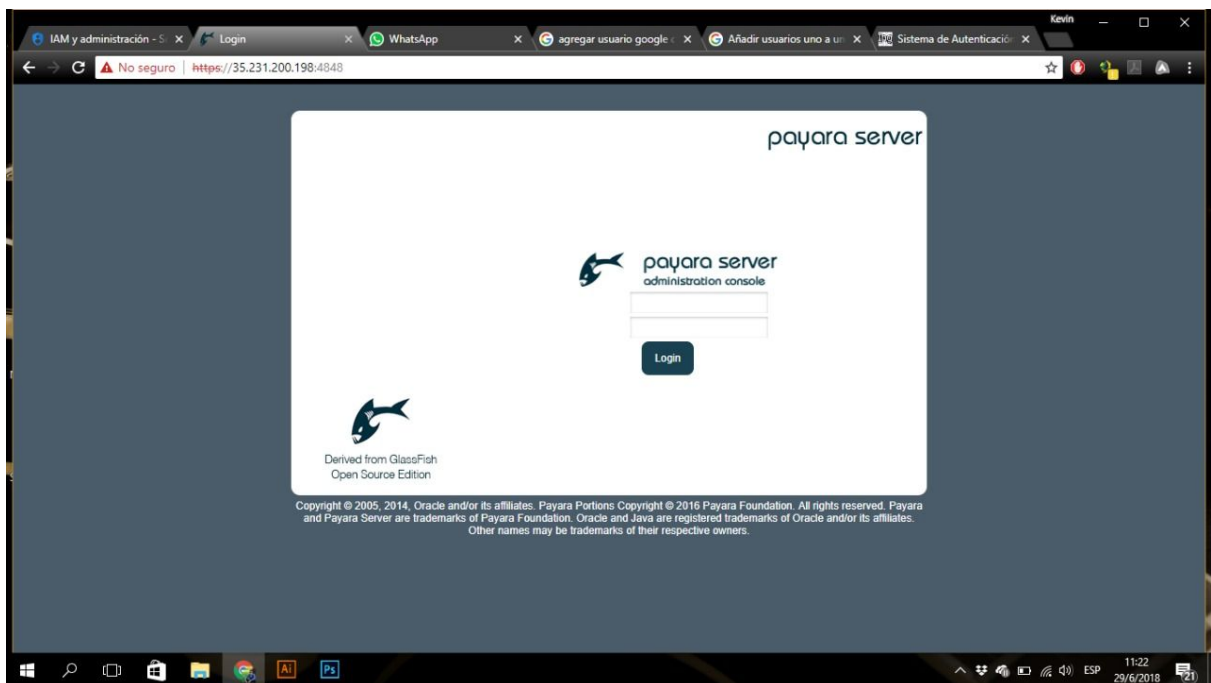
```
wget  
https://s3-eu-west-1.amazonaws.com/payara.fish/Payara+Downloads/5.182/  
/payara-5.182.zip
```

Descomprimir

```
sudo apt-get install unzip  
sudo unzip payara-5.182.zip  
cd payara5/bin/  
sudo ./asadmin start-domain
```

Habilitar el puerto de administracion y cambiar la contraseña

```
sudo ./asadmin change-admin-password  
sudo ./asadmin --host localhost --port 4848 enable-secure-admin
```



SSH Local

Creamos una llave para poder ejecutar

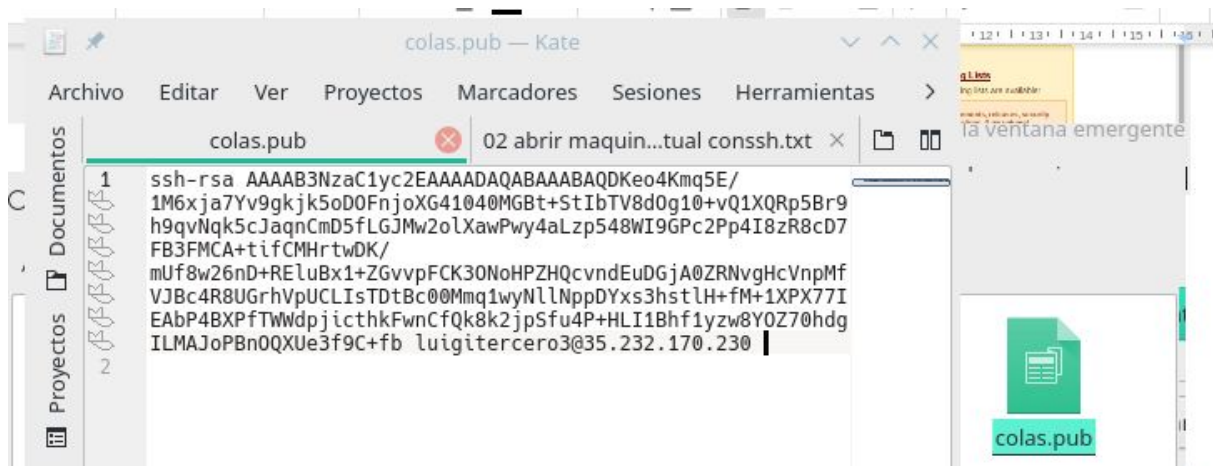
ssh-keygen

Escribir el nombre del archivo

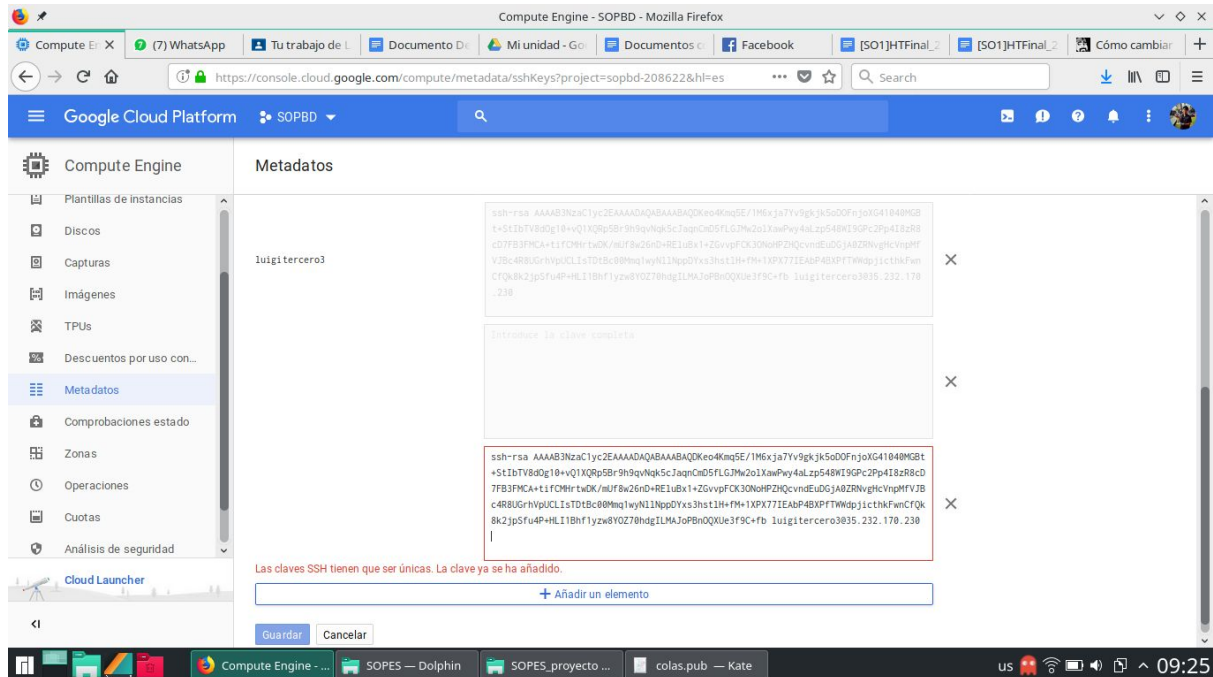
enter 2 veces se crea el archivo en la carpeta actual con extensión .pub

se busca en la máquina virtual que ip pública tiene

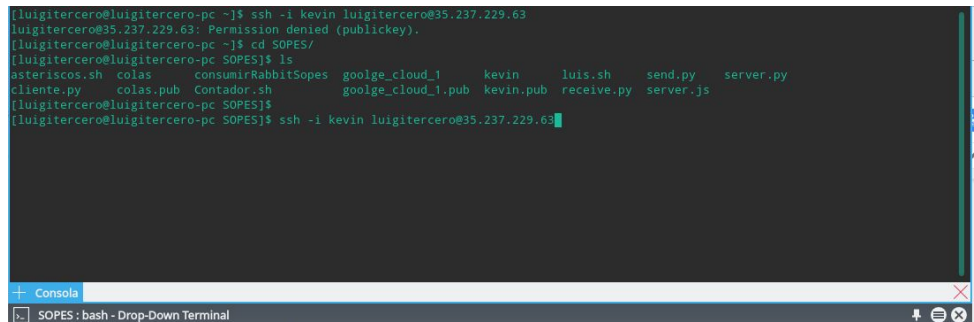
y se coloca como usuario en la última parte luigitercero3@35.203.14.78



En google cloud dirigir a metadata->claves ssh y por último agregar los datos del archivo que se genero de .pub pegarlos y guardar el archivo



En la maquina cliente ssh -i [archivo.pub] [Usuario]@[ipPublica]
ssh -i google_cloud_1 luigitercero3@35.203.14.78



Instalar haproxy

Se agrega al repositorio las nueva librerías

```
sudo add-apt-repository ppa:vbernat/haproxy-1.8
```

Actualizar los repositorios

```
sudo apt-get update
```

Instalar haproxy

```
sudo apt-get install haproxy
```

Consultar versión

```
haproxy -v
```

```
HA-Proxy version 1.8.12-1ppa1~xenial 2018/06/27  
Copyright 2000-2018 Willy Tarreau <willy@haproxy.org>
```

Crear como un back up de los archivos por defecto

```
sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.real.cfg
```

En bin agregar el balanceador de carga

```
vim /etc/haproxy/haproxy.cfg
```

frontend http_front

```
bind *:80
```

```
stats uri /haproxy?stats
```

```
default_backend http_rear
```

backend http_rear

```
balance roundrobin
```

```
server wbs1.com 192.168.5.155:80 check #ip de valanceo
```

```
server wbs2.com 192.168.5.147:80 check #ip de valanceo
```

salir ctrl + c :wq

Reiniciar el servidor

```
sudo systemctl restart haproxy
```

Consultar página de balanceo

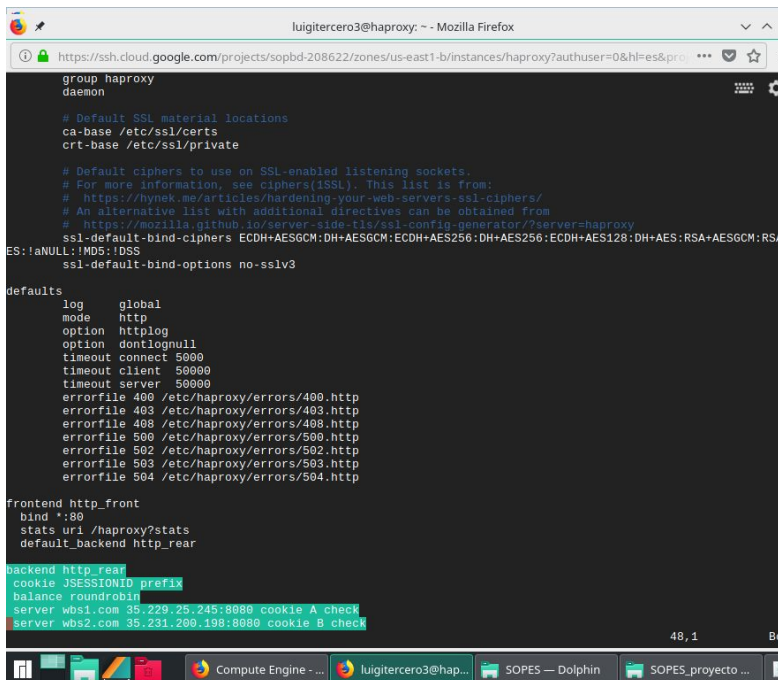
```
http://35.203.14.78/haproxy?stats
```

Agregar password

```
listen stats
bind *:8800
stats enable
stats uri /
stats realm Haproxy\ Statistics
stats auth admin:12345
```

<https://www.youtube.com/watch?v=Ay8jOdu3nK8>

Variable de sesión y cache



```
group haproxy
daemon

# Default SSL material locations
ca-base /etc/ssl/certs
crt-base /etc/ssl/private

# Default ciphers to use on SSL-enabled listening sockets.
# For more information, see ciphers(1SSL). This list is from:
# https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
# An alternative list with additional directives can be obtained from
# https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy
ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA
ES:!aNULL:!MD5:!DSS
ssl-default-bind-options no-sslv3

defaults
log global
mode http
option httplog
option dontlognull
timeout connect 5000
timeout client 50000
timeout server 50000
errorfile 400 /etc/haproxy/errors/400.http
errorfile 403 /etc/haproxy/errors/403.http
errorfile 408 /etc/haproxy/errors/408.http
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http

frontend http_front
bind *:80
stats uri /haproxy?stats
default_backend http_rear

backend http_rear
cookie JSESSIONID prefix
balance roundrobin
server wbs1.com 35.229.25.245:8080 cookie A check
server wbs2.com 35.231.200.198:8080 cookie B check
```

Statistics Report for HAProxy - Mozilla Firefox

35.229.32.168/haproxy?stats

HAProxy version 1.8.12-1ppa1-xenial, released 2018/06/27

Statistics Report for pid 29806

> General process information

pid = 29806 (process #1, nbproc = 1, nbthread = 1)
 uptime = 2d 6h52m33s
 system limits: memmax = unlimited; ulimit-n = 4033
 maxsock = 4033; maxconn = 2000; maxpipes = 0
 current conns = 5; current pipes = 0/0; conn rate = 0/sec
 Running tasks: 1/11; idle = 100 %

active UP
 active UP, going down
 active DOWN, going up
 active or backup DOWN
 active or backup DOWN for maintenance (MAINT)
 active or backup SOFT STOPPED for maintenance

backup UP
 backup UP, going down
 backup DOWN, going up
 not checked
 Note: "NOLB/DRAIN" = UP with load-balancing disabled.

Display option:
 Scope:
 Hide DOWN servers
 Refresh now
 CSV export

External resources:
 Primary site
 Updates (v1.8)
 Online manual

http_front																														
	Queue			Session rate			Sessions				Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrtle			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn										Resp	Retr	Redis
Frontend	0	6	-	5	13	2 000	228						202 185	6 966 538	0	0	9								OPEN					

http_rear																												
	Queue			Session rate			Sessions				Bytes		Denied		Errors		Warnings		Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtime	Thrtle	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn										Resp
ws1.com	0	0	-	0	31	0	4	-	377	91	19m19s	171 763	5 490 634	0	0	1	0	0	0	2d6h UP	L40k in 1ms	1	Y	-	0	0	0s	-
ws2.com	0	0	-	0	17	0	2	-	126	91	15s	30 363	1 474 212	0	0	0	0	0	0	2d6h UP	L40k in 1ms	1	Y	-	1	0	0s	-
Backend	0	0	-	0	31	0	4	200	503	182	15s	202 126	6 964 846	0	0	0	1	0	0	2d6h UP		2	2	0	0	0s	-	

us 09:37

Instalación Cassandra en Ubuntu 14.04

<https://hostpresto.com/community/tutorials/how-to-install-apache-cassandra-on-ubuntu-14-04/>

Actualizar base de datos

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

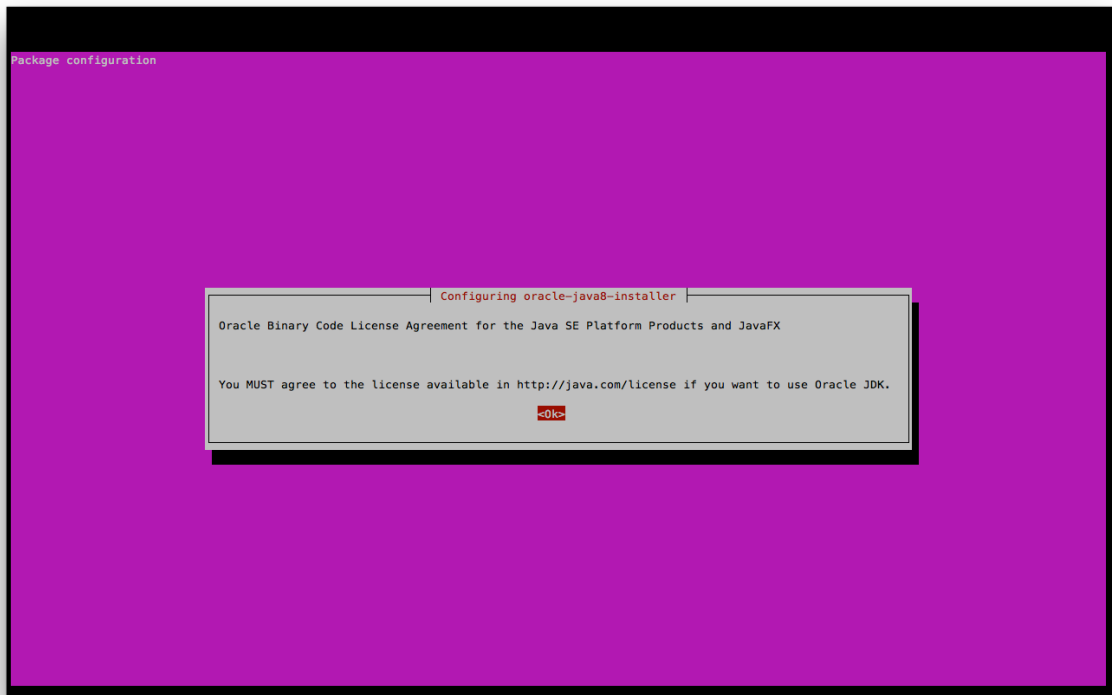
Instalar JDK 8

```
$ sudo add-apt-repository ppa:webupd8team/java
...
Press [ENTER] to continue or ctrl-c to cancel adding it
...
OK
```

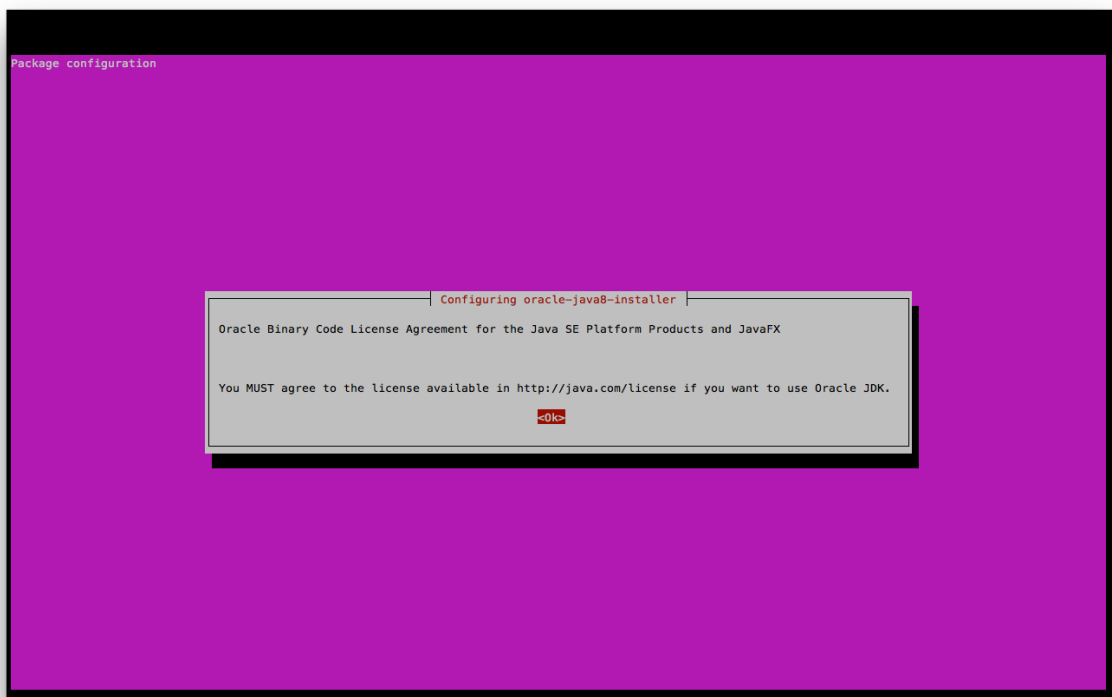
```
$ sudo apt-get update
```



```
$ sudo apt-get -y install oracle-java8-installer
```



Accepting **Oracle Binary Code Licence Terms**. Choose Yes



```
$ java -version
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)
```

Install Apache Cassandra

```
$ curl -L http://debian.datastax.com/debian/repo_key | sudo apt-key add -
```

```
$ echo "deb http://debian.datastax.com/community stable main" | sudo tee -a
/etc/apt/sources.list.d/cassandra.sources.list
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install dsc22=2.2.3-1 cassandra=2.2.3
```

```
$ sudo apt-get install cassandra-tools=2.2.3
```

```
$ sudo service cassandra status
```

```
Datacenter: datacenter1
=====
```

```
Status=Up/Down
```

```
/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns	Host ID	Rack
UN	127.0.0.1	179.29 KB	256	?	7cd1bdc4-8bfa-49d9-a453-e0cf83bf956f	rack1

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless

Let's try connecting to Cassandra server using cqlsh. You can use the command below

```
$ cqlsh
```

```
Connected to Test Cluster at 127.0.0.1:9042.
```

```
[cqlsh 5.0.1 | Cassandra 2.2.3 | CQL spec 3.3.1 | Native protocol v4]
```

```
Use HELP for help.
```

```
cqlsh> quit
```

Redis

Instalar Redis

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

```
$ sudo apt-get -y install redis-server
```

```
$ sudo service redis-server status
```

Configurar Redis

```
$ sudo netstat -nptu | grep LISTEN
```

```
bind 127.0.0.1
```

```
$ sudo service redis-server restart
```

Configurar seguridad en redis

```
abrir /etc/redis/redis.conf
```

find the line below

```
# requirepass foobared
```

Replace foobared the line above with your own password. You can also use a fully random password like the line below

```
requirepass nl6Cq8mthJrrXbqIDqLaPgtFkeq12zqB7Sb5j5UJ
```

```
$ sudo service redis-server restart  
Stopping redis-server: redis-server.  
Starting redis-server: redis-server
```

```
$ redis-cli
```

```
127.0.0.1:6379> PING
```

```
127.0.0.1:6379> AUTH nl6Cq8mthJrrXbqIDqLaPgtFkeq12zqB7Sb5j5UJ  
OK
```

```
127.0.0.1:6379> PING  
PONG  
127.0.0.1:6379>
```

Redis

empaquetado <https://docs.bitnami.com/google/infrastructure/redis/>

Google Cloud Platform - SOPBD

Compute Engine

Instancias de VM

La instancia 'colitas' está infrautilizada

Filtrar las instancias de VM

Nombre	Zona
colitas	us-east1-b
haproxy	us-east1-b
luis201408606	us-east1-b

redis

- redis Instance: sopbd-208523
- redisdb-vm Instance: sopbd-208523
- redis Disk: sopbd-208523
- redisdb-vm Disk: sopbd-208523
- API Cloud Memorystore for Redis API
- Redis 3 Google Click to Deploy containers
- Redis 4 Google Click to Deploy containers
- Redis Certified by Bitnami Bitnami
- Redis High Availability Bitnami
- Secured Redis on Red Hat Enterprise Linux 7 Cognosys Inc.

Cloud Launcher

Windows taskbar: 0:13 04/07/2018

Google Cloud Platform - SOPBD

Redis Certified by Bitnami

Bitnami

Coste estimado: 4,28 \$/mes | Más de 100 despliegues recientes

Database software from the leading publisher

EJECUTAR EN COMPUTE ENGINE

Se ejecuta en Google Compute Engine

Tipo Máquinas virtuales VM única

Última actualización 3/7/18 23:49

Categoría Bases de datos Herramientas desarrollo

Visión general

Redis is an open source, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets and sorted sets.

Why use Bitnami Certified Apps?

Bitnami certifies that our images are secure, up-to-date, and packaged using industry best practices.

With Bitnami you can trust what's in the app you're launching. We monitor all components and libraries for vulnerabilities, outdated components, and application updates. When one is reported, we update and release every affected listing within a couple days at most.

Más información

Windows taskbar: 0:14 04/07/2018

(7) WhatsApp x How to install and config x [SO1]HTFinal_201408606 x Cloud Launcher - SOPBD x

Secure | https://console.cloud.google.com/launcher/config/bitnami-launchpad/redis?src=console&project=sopbd-208622&hl=es

Google Cloud Platform SOPBD

Nuevo despliegue de Redis Certified by Bitnami

Deployment name

Zone

us-central1-f

Machine type

micro (1 vCPU ...)

0,6 GB de memoria

Personalizar

Boot Disk

Boot disk type

Standard Persistent Disk

Boot disk size in GB

10

Networking

Network name

default

Subnetwork name

default

Más

Visión general de Redis Certified by Bitnami

Solución proporcionada por Bitnami

4,28 \$ al mes (estimación)

Tarifa por horas efectiva: 0,006 \$ (730 horas al mes)

Detalles

Software

Sistema operativo	Debian (9)
Software	OpenSSL (1.0.2o)
	Redis (4.0.10)

Condiciones de Servicio

El software o el servicio que estás a punto de usar no es un producto de Google. Si instalas el servicio o accedes a él, estarás aceptando cumplir las [Condiciones del Servicio de Bitnami](#) y las [Condiciones de Servicio de Google Cloud Launcher](#), así como las condiciones de cualquier licencia de software de terceros relacionada con el software o el servicio. Revisa detenidamente estas licencias, y comprueba todas las obligaciones que se te aplicarán con respecto al software o al servicio. En la medida en que una licencia de software de código abierto relacionada con el software o el servicio sustituya expresamente las Condiciones de Servicio de Google Cloud Launcher, dicha licencia de software de código abierto registrará tu uso del software o servicio.

Google suministra este software o servicio "tal cual", y cualquier tipo de asistencia con respecto a este software o servicio la proporcionará Bitnami conforme a sus Condiciones de Servicio.

(7) WhatsApp x How to install and config x [SO1]HTFinal_201408606 x Deployment Manager - x

Secure | https://console.cloud.google.com/dm/deployments/details/redis-1?src=launcher&project=sopbd-208622&hl=es

Google Cloud Platform SOPBD

Deployment Manager

Despliegues

Registro de tipos

redis-1

DETENER ELIMINAR

redis-1 se está desplegando

Visión general: redis-1

- redis-jinja
 - redis-vm-tmpl vm_instance.py
 - redis-1-vm instancia de VM
 - generated-password-0 password.py
 - software-status software_status.py
 - redis-1-config configuración
 - redis-1-software recurso de espera de configuración
 - software-status-script software_status_script.py

Redis Certified by Bitnami

Solución proporcionada por Bitnami

Admin user	user
Admin password (Temporary)	Pendiente
Instance	Pendiente
Instance zone	Pendiente
Instance machine type	Pendiente

Más información sobre el software

Empieza a utilizar Redis Certified by Bitnami

Podrás usar Redis Certified by Bitnami cuando termine de implementarse.

Documentación

[Getting Started](#)

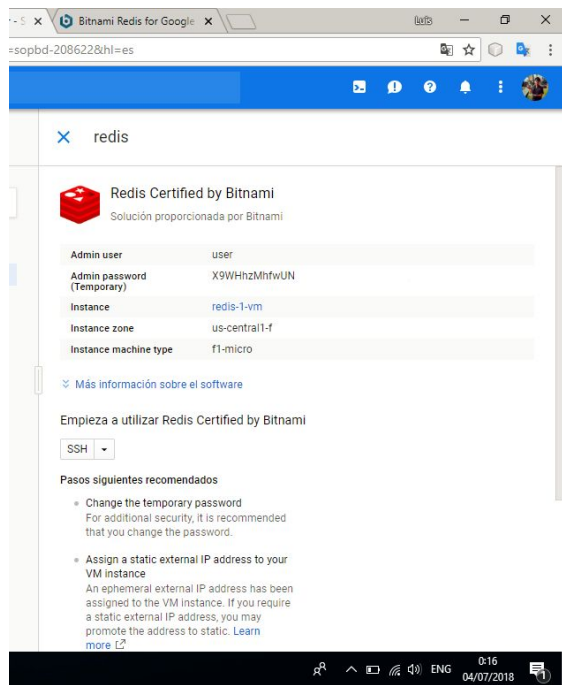
Get started with Bitnami Redis Stack.

Asistencia

Bitnami provides technical support for installation and setup issues through our [support center](#). [Ir al servicio de asistencia de Bitnami](#)

Propiedades de la plantilla

Más



se puede entrar como máquina virtual

RabbitMQ

<https://tecadmin.net/install-rabbitmq-server-on-ubuntu/>

Install Erlang

```
wget https://packages.erlang-solutions.com/erlang-solutions_1.0_all.deb
```

```
sudo dpkg -i erlang-solutions_1.0_all.deb
```

```
sudo apt-get update
```

```
sudo apt-get install erlang erlang-nox
```

Manage RabbitMQ Service

Using Init

```
sudo update-rc.d rabbitmq-server defaults
```

```
sudo service rabbitmq-server start
```

```
sudo service rabbitmq-server stop
```

Using Systemctl

```
sudo systemctl enable rabbitmq-server
```

```
sudo systemctl start rabbitmq-server
```

```
sudo systemctl stop rabbitmq-server
```

Create Admin User in RabbitMQ

```
sudo rabbitmqctl add_user admin password
```

```
sudo rabbitmqctl set_user_tags admin administrator
```

```
sudo rabbitmqctl set_permissions -p / admin ".*" ".*" ".*"
```


Setup RabbitMQ Web Management Console

```
sudo rabbitmq-plugins enable rabbitmq_management
```



APACHE BENCHMARK windows 10

Apache Benchmark (también conocido como ab) es una herramienta para comparar el servidor web HTTP. Se recomienda probar el rendimiento de su servidor web antes de cambiarlo al entorno de producción. Se usa estas herramientas para realizar pruebas comparativas y hacer una prueba de estrés a nuestro servidor de desarrollo antes de que se active.

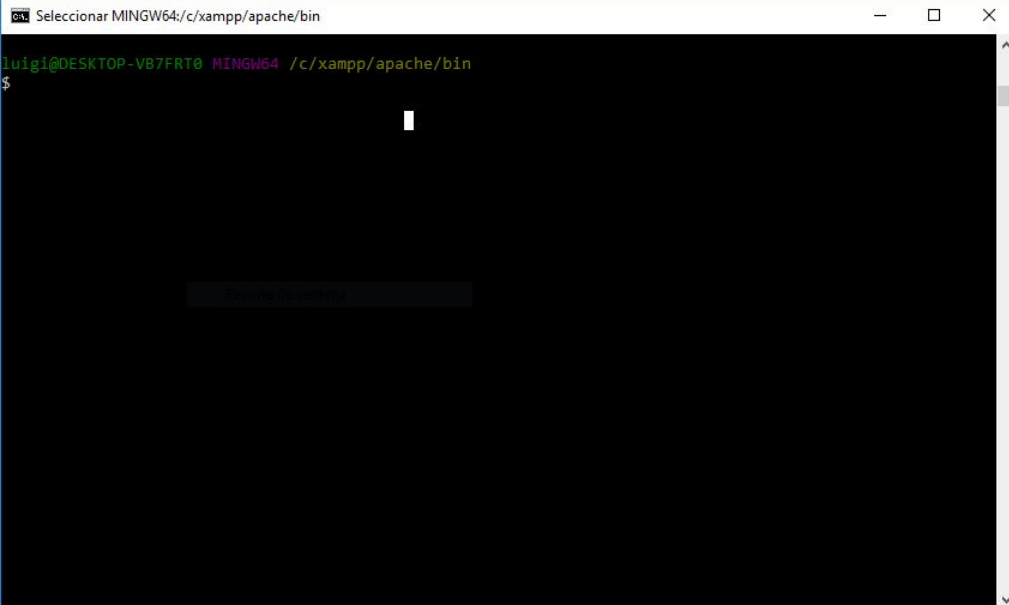
Intalacion de xamp :

<http://www.mclibre.org/consultar/php/otros/xampp-instalacion-windows.html>

antes de empezar se debe preparar el entorno, estos son los requerimientos:

- La URL es accesible a través de dominio público o IP: <127.0.0.1:8000/>
- Espere el número de clientes que su servidor esté listo para servir: 50 usuarios simultáneos / segundos
- Espere el número de solicitudes por cliente que su servidor esté listo para servir: 10 solicitudes / usuario / segundos
- Gnuplot <http://www.gnuplot.info/download.html>

En una terminal dirigirse a la carpeta de C:/xampp/apache/bin

A screenshot of a terminal window titled 'Seleccionar MINGW64:/c:/xampp/apache/bin'. The prompt shows the user 'luigi@DESKTOP-VB7FRT0' in a 'MINGW64' environment at the directory '/c:/xampp/apache/bin'. The prompt is '\$' followed by a cursor. The terminal background is black with green text for the prompt and a white cursor.

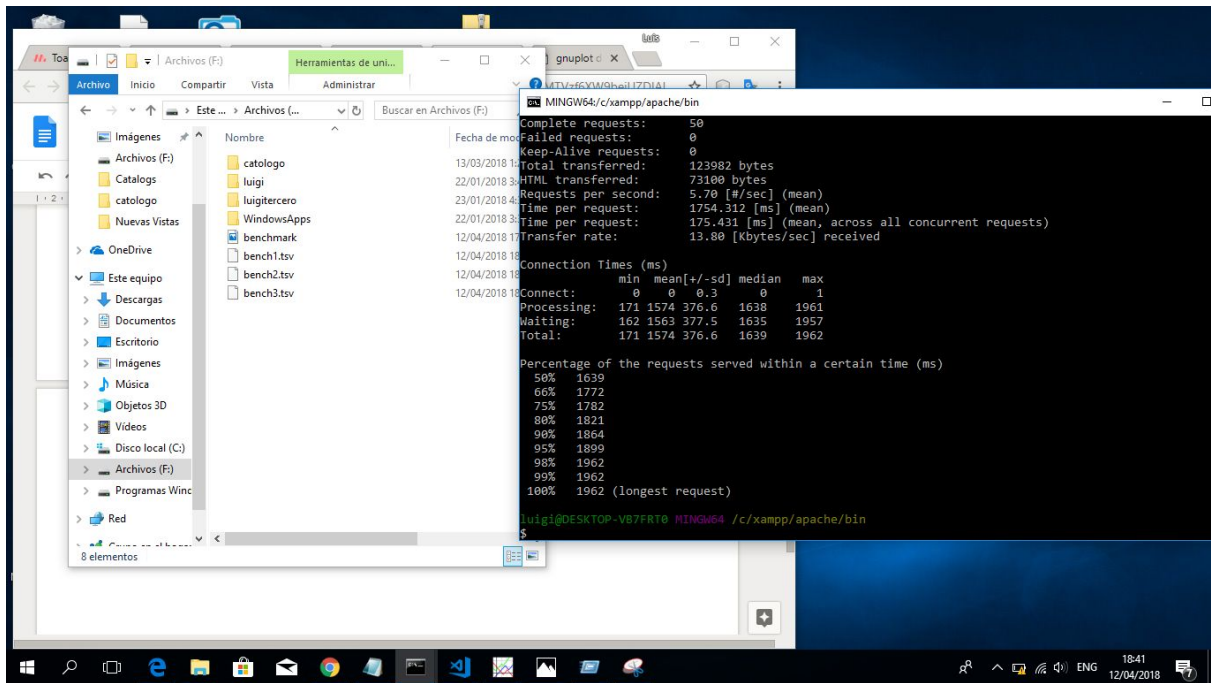
para empezar con las pruebas

`./ab -c <cantidad de clientes> -n <cantidad de usuarios simultaneos>
-k -g <Dirección de archivo> <deConsulta>`

`./ab -c 100 -n 5000 -k -g f:/bench1.tsv 127.0.0.1:8000/`

`./ab -c 10 -n 50 -k -g f:/bench2.tsv 127.0.0.1:8000/`

debemos verificar los archivos se hayan creado



Nos dirigimos a gnuplot

salida es un png

set terminal png

guardar el archivo en"

set output "F:\benchmark.png"

título de la gráfica

set title "Benchmark for practica 1"

#tam;o ancho

set size 1,1

deshabilitar el y

set grid y

título de eje de x

set xlabel "Request"

#título de eje de y

set ylabel "Response Time (ms)"

plotetar los 3 archivos con suabizador

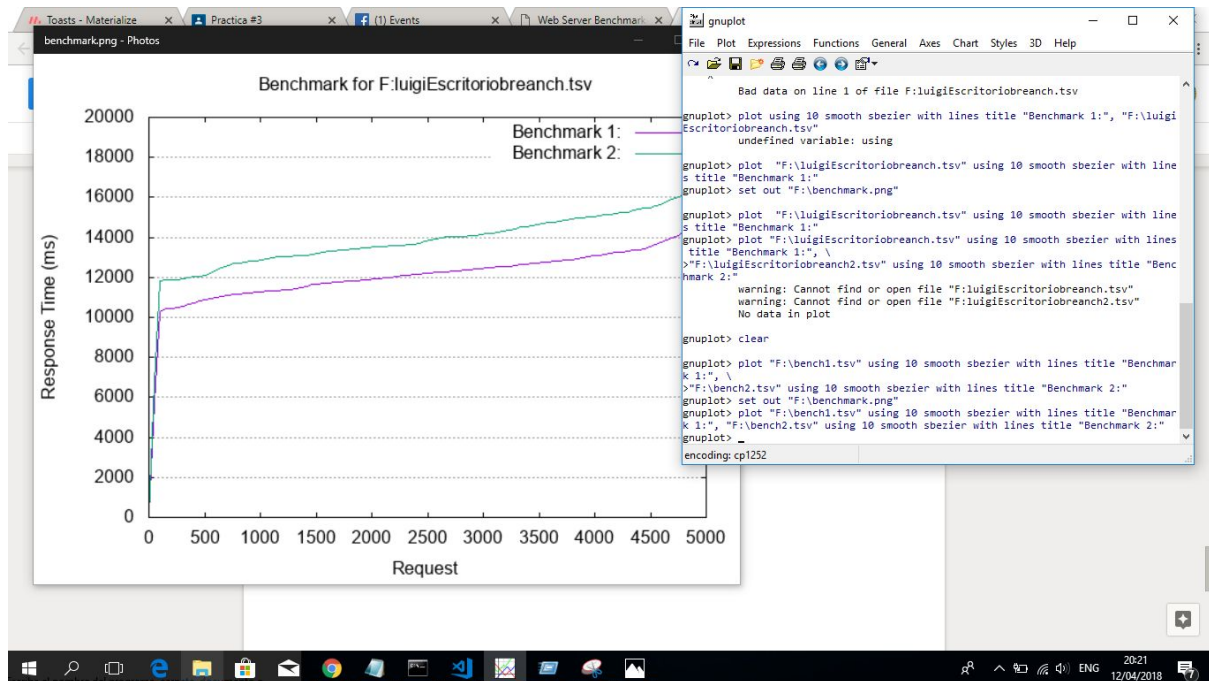
plot "F:\bench1.tsv" using 10 smooth sbezier with lines title "Benchmark 1:", "F:\bench2.tsv"

using 10 smooth sbezier with lines title "Benchmark 2:"

Nota:

es posible crear un script con las sentencias en un archivo ejemplo:
gnuplot benchmark.tpl

se crea la imagen



si usted observa los archivos tsv creados por ab, se muestra las columnas **ctime**, **dtime**, **ttime** and **wait**. que son definidas

ctime: tiempo de coneccion

dtime: tiempo procesando

ttime: tiempo total

wait: tiempo de espera

Implementación de arquitectura

Cliente

El cliente es el que podrá visualizar las paginas, ingresando al servidor del HaProxy

LB(HaProxy)

Este servidor es un load balancer o balanceador de carga, debe de ser configurado con Haproxy este debe de recibir el trabajo y distribuirlo a los servidores Payara y Tomcat usando el algoritmo Round Robin.

Servidor 1 - Tomcat:

Este servidor es el que tiene una instancia de la aplicación. es una máquina virtual con un servidor Tomcat. Este estará contenido en una máquina virtual.

Servidor 2 - Payara:

Este servidor es el que tiene una segunda instancia de la aplicación. es una máquina virtual con un servidor Payara. El modularidad de este proyecto recae en la reusabilidad de la aplicación, es decir, una vez hecha la aplicación

Servidor de cola de mensaje:

este servidor contendrá una cola de mensajes, la cual se comunicará con la base de datos a utilizar (Redis) para procesar la lectura y escritura de datos y también contendrá otra cola de mensajes, la cual se comunicará con la base de datos (Cassandra), Es decir existirán dos colas de mensajes.

Rabbit como servidor de colas, verificar la documentación de Rabbit para ser el más adecuado a utilizar,

Redis:

base de datos, esta base de datos almacenara lo siguiente

- Usuario
- Comentario realizado por el mismo usuario.

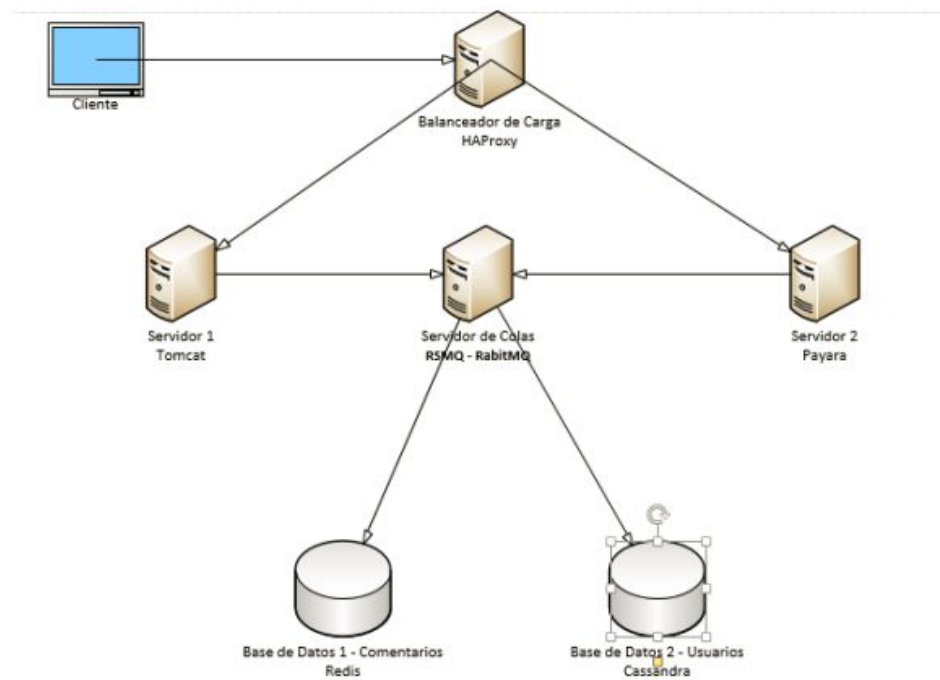
Cassandra:

Base de datos para el control de usuarios, La aplicación a realizar deberá de ser un sistema de autenticación (Login) así como el registro de usuarios con un formulario de registro. Además, se deberá de poder ingresar a una interfaz para la consulta de usuarios registrados, siempre y cuando el usuario este autenticado. También se podrá visualizar la información de usuarios registrados por medio de una búsqueda.

De cada usuario se deberá de registrar:

- Nombre de usuario
- Contraseña
- Nombre
- Apellido
- Fecha de registro (con hora, minutos y segundos)

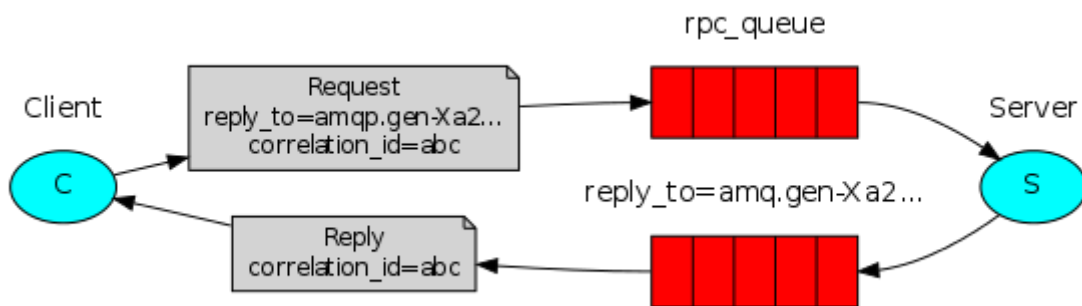
Arquitectura



Uso de Rabbitmq con colas

[RPC](#)

[Request/reply pattern](#)



se utilizó este modelo de colas, por que es ideal para la respuesta que necesita cada cliente con el servidor

se utilizó como servidor el lenguaje de de

Servidor

python

servidor RbbitMQ con Redis

<https://github.com/luigitercero/consumirRabbitSopes/blob/master/mensajes.py>

Explicación

libreria **pika** esta es utilizada para la conexion remota con los servidores

libreria **redis** se utiliza para obtener los metodos de la base de datos de redis

```
def guardarDatos(mensaje):
```

este es un metodo para guardar los mensajes de los usuario obtiene una cadena y retorna un 1 para indicar que se guardo el mensaje

```
def retornar():
```

este es un método que devuelve la consulta de todos los comentarios guardados en la base de datos


```
def on_request(ch, method, props, body):
```

Este procedimiento escucha con regularidad al servidor donde se encuentran las colas en rabbit mq

servidor RbbitMQ con Cassadra

<https://github.com/luigitercero/consumirRabbitSopes/blob/master/server.py>

Explicación

```
def actualizarUser(datos):
```

este metodo acutaliza la base de datos retorna un 5 para avisar que se actualizo

```
def insertarBd(datos):
```

este metotodo sirve para el uso de insercciones en la base de datos

```
def accederUser(datos):
```

este metodo sirve para que el usuario pueda loguearse, devuelve un 1 para ingresar

```
accion(body):
```

es un metodo para parasear la cadena que viene de rabbit y elegir lo que pueda venir

```
def veruser(datos):
```

este es un metodo para ver los datos del usuario

```
def on_request(ch, method, props, body):
```

Este procedimiento escucha con regularidad al servidor donde se encuentran las colas en rabbit mq

Cliente

java

https://github.com/kevinmoran100/Cliente_sopes1/tree/master/src/java

Comunicación con rabbitmq

Para la comunicación con rabbitmq se crearon dos clases una para la comunicación con cada una de las colas.

[RPCClient_Cassandra.java](#): se comunica con la cola de rabbit para la comunicación la base de datos Cassandra que guarda los usuarios.

[RPCClient_Redis.java](#): se comunica con la cola de rabbit para la comunicación con la base de datos Redis que guardan los mensajes.

En ambas clases se tiene el siguiente método

```
public String call(String message):
```

Es un método que sirve para enviar el mensaje a determinada cola y a su vez crea una cola temporal que servirá para recibir la respuesta.

Servlets

Para la comunicación con las colas y las paginas web se manejaron servlets en java.

[ServletCuenta.java](#): Este servlet se carga al solicitar la dirección “/cuenta” y su función es solicitar en la cola la información sobre el usuario que tiene una sesión abierta actualmente y luego carga la información en la página web.

pruebas de Estres

