

Apache Usergrid\_

Ramiro Belluardo

Julio 2016



*apache*  
**usergrid\_**

# 1- Introduccion

Usergrid es una aplicación de código abierto compuesto y distribuido por una base de datos NoSql, la capa de aplicación y la capa de cliente con el SDK para los desarrolladores que buscan construir rápidamente web y/o aplicaciones móviles. Proporciona servicios elementales (Registro de usuarios y de gestión, almacenamiento de datos, almacenamiento de archivos, colas) y funciones de recuperación (búsqueda de texto completo, búsqueda de geolocalización)

Es un sistema multi-usuario diseñada para su implementación en entornos de nube pública (como Amazon Web Services, Rackspace, etc.) o para funcionar en infraestructuras de servidores tradicionales, de modo que cualquier persona puede ejecutar su propia implementación privada BaaS.

Para los arquitectos y equipos de back-end, tiene como objetivo proporcionar una solución distribuida, fácilmente ampliable, operativamente predecible y altamente escalable. Para los desarrolladores de aplicaciones para usuario, tiene por objeto simplificar el proceso de desarrollo por lo que les permite construir y operar aplicaciones móviles y web sin necesidad de conocimientos de back-end rápidamente.

En este proyecto explicaremos la instalación de Usergrid en un sistema operativo Linux version 14.04 con una memoria de 8Gb.

## 2-Requisitos Previos

- Linux o un sistema tipo UNIX
- **Java JDK 8** --> <http://tecadmin.net/install-oracle-java-8-jdk-8-ubuntu-via-ppa/>
- **Apache Tomcat 7** --> <https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-7-on-ubuntu-14-04-via-apt-get>
- **Apache Cassandra 1.2.1** --> <https://archive.apache.org/dist/cassandra/1.2.19/>
- **Elasticsearch 1.4** ---> <https://www.elastic.co/downloads/past-releases/elasticsearch-1-4-2>

## 3-Instalacion

Descargamos el usergrid 2.1 del siguiente link:

<http://mirrors.dcarsat.com.ar/apache/usergrid/usergrid-2/v2.1.0>

Descomprimos el archivo .tar y obtendremos lo siguiente

```
+-- apache-usergrid-2.1.0
|
|+-- LICENSE
|
|+-- NOTICE
|
|+-- CHANGELOG
|
|+-- stack
| |
| | + ROOT.war
|
|+-- portal
| |
| | +-- dist
| |   |
| |   + usergrid-portal.tar
|
|+-- sdks
| |
| | +-- html5-javascript (JavaScript SDK and source)
| |
| | +-- java (Java SDK and source)
```

Los archivos que nos interesan son usergrid-portal.tar y ROOT.war

### La implementación de la pila Usergrid

El Usergrid Stack es una aplicación web Java EE que se ejecuta en Tomcat, utiliza la base de datos Cassandra para el almacenamiento y el motor de búsqueda Elasticsearch para las consultas.

Inicializamos cassandra para esto nos paramos en la carpeta donde lo descargamos

Y ejecutamos

```
sudo ./bin/cassandra -f
```

Inicializamos elasticsearch para esto nos paramos en la carpeta donde lo descargamos

Y ejecutamos

```
sudo ./bin/elasticsearch
```

## Configuración de Tomcat

El Usergrid Pila está contenida en un archivo denominado ROOT.war, un WAR estándar Java EE listo para el despliegue de Tomcat.

## Configurar Usergrid pila y tala

Se debe crear un archivo de propiedades Usergrid llamado `usergrid-deployment.properties`. Este archivo dice cómo comunicarse con Cassandra y Elasticsearch, y cómo formar URL utilizando el nombre de host que desea utilizar para Usergrid. Hay muchas propiedades que se pueden establecer para configurar Usergrid.

Una vez creado el archivo de propiedades Usergrid, lo colocan en el directorio lib de Tomcat. (Para esto acceder como root) Usualmente situado en `/usr/share/tomcat7/lib`.

# Usergrid-deployment.properties

```
usergrid.cluster_name=${USERGRID_CLUSTER_NAME}

cassandra.url=${CASSANDRA_HOSTS}
cassanrda.cluster=${CASSANDRA_CLUSTER_NAME}

elasticsearch.cluster_name=${ELASTICSEARCH_CLUSTER_NAME}
elasticsearch.hosts=${ELASTIC_SEARCH_HOSTS}

#####
# Admin and test user setup

usergrid.sysadmin.login.allowed=true
usergrid.sysadmin.login.name=superuser
usergrid.sysadmin.login.password=${SUPER_USER_PASSWORD}
usergrid.sysadmin.login.email=${SUPER_USER_EMAIL}

usergrid.sysadmin.email=${SUPER_USER_EMAIL}
usergrid.sysadmin.approve.users=true
usergrid.sysadmin.approve.organizations=true

# Base mailer account - default for all outgoing messages
usergrid.management.mailer=Admin <${SUPER_USER_EMAIL}>

usergrid.setup-test-account=true
usergrid.test-account.app=test-app
usergrid.test-account.organization=test-organization
usergrid.test-account.admin-user.username=test
usergrid.test-account.admin-user.name=Test User
usergrid.test-account.admin-user.email=${TEST_ADMIN_USER_EMAIL}
usergrid.test-account.admin-user.password=${TEST_ADMIN_USER_PASSWORD}

#####
# Auto-confirm and sign-up notifications settings

usergrid.management.admin_users_require_confirmation=false
usergrid.management.admin_users_require_activation=false

usergrid.management.organizations_require_activation=false
usergrid.management.notify_sysadmin_of_new_organizations=true
usergrid.management.notify_sysadmin_of_new_admin_users=true

#####
# URLs

# Redirect path when request come in for TLD
usergrid.redirect_root=${BASEURL}/status

usergrid.view.management.organizations.organization.activate=${BASEURL}/accounts/welcome
```

usergrid.view.management.organizations.organization.confirm=\${BASEURL}/accounts/welcome

usergrid.view.management.users.user.activate=\${BASEURL}/accounts/welcome

usergrid.view.management.users.user.confirm=\${BASEURL}/accounts/welcome

usergrid.admin.confirmation.url=\${BASEURL}/management/users/%s/confirm

usergrid.user.confirmation.url=\${BASEURL}/%s/%s/users/%s/confirm

usergrid.organization.activation.url=\${BASEURL}/management/organizations/%s/activate

usergrid.admin.activation.url=\${BASEURL}/management/users/%s/activate

usergrid.user.activation.url=\${BASEURL}/%s/%s/users/%s/activate

usergrid.admin.resetpw.url=\${BASEURL}/management/users/%s/resetpw

usergrid.user.resetpw.url=\${BASEURL}/%s/%s/users/%s/resetpw

Reemplazamos en cada valor que esta en la tabla por la descripcion

Value	Description
<b>BASEURL</b>	El url donde instalamos usergrid . Ej:http://localhost:8080
<b>USERGRID_CLUSTER_NAME</b>	Nombre de uso en la instalación.
<b>CASSANDRA_CLUSTER_NAME</b>	Nombre de cassandra cluster.  Ubicado en cassandra.yaml
<b>CASSANDRA_HOSTS</b>	Cassandra host. por default http://localhost:9160
<b>ELASTICSEARCH_CLUSTER_NAME</b>	Nombre del cluster de elasticsearch.  Ubicado en elasticsearch.yaml.
<b>ELASTICSEARCH_HOSTS</b>	Elasticsearch host. por default <code>localhost:9300</code>
<b>SUPER_USER_EMAIL</b>	Email address of person responsible for the superuser account.

<b>SUPER_USER_PASSWORD</b>	contraseña de su cuenta de superusuario
<b>TEST_ADMIN_USER_EMAIL</b>	Si <code>usergrid.setup-test-account=true</code> , tienes que poner un email valido ya que usergrid verifica
<b>TEST_ADMIN_USER_PASSWORD</b>	contraseña de su cuenta de test

## Creacion de archivo usergrid-custom.properties

Se debe crear un archivo de propiedades Usergrid llamado `usergrid-custom.properties`.

Una vez creado el archivo lo colocan en el directorio lib de Tomcat. En un sistema Linux, este directorio es, probablemente, situado en `/usr/share/tomcat7/lib`.

### usergrid-custom.properties

```
# Admin and test user setup

usergrid.sysadmin.login.allowed=true
usergrid.sysadmin.login.name=superuser
usergrid.sysadmin.login.password=${SUPER_USER_PASSWORD}
usergrid.sysadmin.login.email=${SUPER_USER_EMAIL}

usergrid.sysadmin.email=${SUPER_USER_EMAIL}
usergrid.sysadmin.approve.users=true
usergrid.sysadmin.approve.organizations=true
```

Completan con lo mismo que pusieron en el archivo Usergrid-deployment.propety

## configurar el registro

1. Crear el archivo `log4j.properties`

2. Añadir propiedad `-d` del sistema para Tomcat. De manera que Tomcat pueda encontrar el archivo de propiedades

## Log4j.properties

```
# output messages into a rolling log file as well as stdout
log4j.rootLogger=ERROR,stdout

# stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %p (%t) [%c] - %m%n

log4j.logger.org.apache.usergrid=INFO

log4j.logger.me.prettyprint.cassandra.hector.TimingLogger=WARN
log4j.logger.org.apache.usergrid.rest.security.AllowAjaxFilter=WARN
log4j.logger.me.prettyprint.hector.api.beans.AbstractComposite=ERROR
```

## Añadir el registro de configuración de Tomcat

Una forma de añadir la propiedad anterior a la puesta en marcha de Tomcat es añadir una línea a un Tomcat `setenv.sh` secuencia de comandos en el directorio `bin` de Tomcat. Si no existe ese archivo, crearlo.

Por ejemplo, si su archivo de propiedades se encuentra en `/usr/share/tomcat7/lib/log4j.properties`, a continuación, deberá añadir la siguiente línea a `setenv.sh`:

```
JAVA_OPTS exportación = "-Dlog4j.configuration = file:///usr/share/tomcat7/lib/log4j.properties"
```

Si el archivo ya existe y ya se establece la variable `JAVA_OPTS`, entonces tendrá que

## implementar ROOT.war en Tomcat

El siguiente paso es implementar el software Usergrid Pila de Tomcat. Para eso, colocamos la pila Usergrid `ROOT.war` en el directorio `webapps` de Tomcats, a continuación, reinicie Tomcat.

Usted puede ver el registro de errores de Tomcat con el siguiente comando

```
tail -f /var/log/tomcat7/catalina.out
```



## ¿Funciona?

Compruebe si Usergrid está en funcionamiento mediante una llamada al estado de punto final. Si el explorador Web se ejecuta en el mismo equipo que Tomcat (Tomcat y es en el puerto 8080), entonces usted puede ir a <http://localhost:8080/status> para ver la página de estado Usergrid.

Si se obtiene un archivo JSON de datos de estado, entonces usted está listo para pasar a la siguiente etapa. Debería ver una respuesta que comienza así:

```
{
  "timestamp" : 1454090178953,
  "duration" : 10,
  "status" : {
    "started" : 1453957327516,
    "uptime" : 132851437,
    "version" : "201601240200-595955dff9ee4a706de9d97b86c5f0636fe24b43",
    "cassandraAvailable" : true,
    "cassandraStatus" : "GREEN",
    "managementApplIndexStatus" : "GREEN",
    "queueDepth" : 0,
    "org.apache.usergrid.count.AbstractBatcher" : {
      "add_invocation" : {
        "type" : "timer",
        "unit" : "microseconds",
        ... etc. ...
```

## *Inicializar la base de datos Usergrid*

A continuación, debe inicializar los sistemas de bases de datos, índices y consulta Usergrid.

Para ello debe emitir una serie de operaciones HTTP utilizando las credenciales de superusuario. Sólo puede hacer esto si Usergrid está configurado para permitir el acceso a través de esta propiedad `superused` `usergrid.sysadmin.login.allowed = true` y si se ha utilizado el ejemplo anterior archivo de propiedades, es permitido.

Los tres operación que debe realizar se expresan por el enrollamiento de los comandos de abajo y, por supuesto, tendrá que cambiar la contraseña (“test”) y el usuario para que coincida con la contraseña de superusuario y su superusuario que se establecio en sus propiedades Usergrid .

```
curl -X PUT http://localhost:8080/system/database/setup -u superuser:test
curl -X PUT http://localhost:8080/system/database/bootstrap -u superuser:test
curl -X GET http://localhost:8080/system/superuser/setup -u superuser:test
```

Cuando se emite cada uno de esos comandos enrollamiento, debería ver un mensaje de éxito de esta manera:

```
{
  "Acción" : "Configuración de Cassandra" ,
  "status" : "ok" ,
  "marca de tiempo" : 1454100922067 ,
  "duración" : 374
```

## Desplegando el Portal Usergrid

El Portal Usergrid es una aplicación HTML5 / JavaScript, un montón de archivos estáticos que se pueden implementar en cualquier servidor web, por ejemplo Apache Tomcat o HTTPD.

Para implementar el Portal a un servidor web, se quiere deshacer el tar la `usergrid-portal.tar` archivo en el directorio que sirve como el directorio raíz de sus páginas web.

Por ejemplo, con Tomcat en Linux, puede hacer algo como esto:

```
cp usergrid-portal.tar /var/lib/tomcat7/webapps
cd /usr/share/tomcat7/webapps
tar xf usergrid-portal.tar
```

Entonces es probable que desee cambiar el nombre del directorio de Portal para algo que va a funcionar bien en una dirección URL. Por ejemplo, si desea que su Portal de existir en la ruta `/portal` continuación:

```
mv usergrid-portal.2.0.18 portal
```

Una vez que haya hecho eso hay un paso más. Es necesario configurar el portal para que pueda encontrar la pila Usergrid. Usted lo hace mediante la edición de los `portales / config.js` y cambiar esta línea:

```
Usergrid.overrideUrl = 'http://localhost:8080/';
```

Para establecer el nombre de host que se va a utilizar para su instalación Usergrid.

Iniciar el servidor Web y el Portal debe estar en funcionamiento en [http://localhost:8080 / portal](http://localhost:8080/portal) o donde se ha implementado.

### Complicaciones:

1-tuve problemas a la hora de la instalación baje los programas actualizados y las versiones entre si ya no eran compatibles. Solucion: Respetar las versiones de los programas nombradas arriba!

2-Nunca me devolvió un json la url : <http://localhost:8080/status>

3-Cuando segui la guía paso a paso de la pagina oficial encontraba errores por ej: te decía que para copiar el archivo ROOT.war uses el siguiente comando:

```
cp ROOT.war /usr/share/tomcat7/webapps
```

la carpeta webapps no se encuentra en esa ruta sino en /var/lib/tomcat7

4-Me sigue tirando un error en el log4j.properties

5-Mucha atención con la configuración de los archivos .properties de esto depende la conexión entre los programas .

### FUENTES:

<https://usergrid.apache.org/docs/installation/deployment-guide.html>

