



+



Ismael Santiago Estévez

Fecha: 01/05/2021

Trabajo fin de ciclo

Índice

| | |
|--|----|
| 1. Introducción..... | 3 |
| 2. ¿Qué es juju?..... | 3 |
| 3. Términos de juju..... | 3 |
| 3.1. Charms..... | 3 |
| 3.2. Bootstrap..... | 3 |
| 3.3. Bundle..... | 4 |
| 3.4. Relaciones..... | 4 |
| 3.5. Servicios juju..... | 4 |
| 4. Escenario..... | 4 |
| 5. Caso práctico juju + AWS..... | 5 |
| 5.1. Instalación de juju..... | 5 |
| 5.2. Creación de cuenta AWS..... | 5 |
| 5.3. Conectar juju con aws..... | 6 |
| 5.4. Configuración e instalación charm Mattermost..... | 9 |
| 5.5. Interfaz GUI..... | 10 |
| 5.6. Borrado de aplicación..... | 12 |
| 5.7. Borrado de credenciales..... | 12 |
| 6. Conclusión..... | 14 |
| 7. Bibliografía..... | 14 |

1. INTRODUCCIÓN

En este proyecto vamos a utilizar la herramienta juju junto con amazon web service (AWS). Para realizar el proyecto sera instalar mediante juju aplicaciones en aws. Estas aplicaciones son lanzadas mediante charm que son un conjunto de scripts para desplegar y administrar los servicios que hagan falta para la aplicación que escogamos. También conocer como funciona AWS y como adaptar juju para que todo sea desplegado correctamente.

Una de las aplicaciones que instalaré con juju será Mattermost que es un servicio de chat en línea de código abierto y autohospedable con intercambio de archivos, búsqueda e integraciones. Está diseñado como un chat interno para organizaciones y empresas, y se comercializa principalmente como una alternativa de código abierto para Slack y Microsoft Teams.

2. ¿QUÉ ES JUJU?

juju es una herramienta desarrollada por Canonical que permite desplegar servicios, aplicaciones, etc en la nube y se puede configurar mediante comandos o mediante interfaz web.

Con juju se pueden tener servicio compuestos, por ejemplo, de una base de datos, un servidor de aplicación, un servidor de archivos y un balanceador de carga. Y todos estos servicios juju los necesita un gestor de infraestructura como en este caso vamos a realizarlo en AWS para ordenar los despliegues, también tiene la opción de utilizar contenedores Linux (LXC), que se desplegaría en maquinas virtuales.

3. TÉRMINOS DE JUJU

3.1. Charms

Los charms son conjuntos de scripts para desplegar y administrar servicios. Con el manejo de eventos integrado, pueden declarar interfaces que encajan con otros servicios, por lo que pueden formarse relaciones.

3.2. Bootstrap

Inicializa un entorno juju para que los servicios se puedan implementar. Bootstrap proporciona una nueva máquina en el entorno y ejecuta el servidor de estado juju en esa máquina (también llamado el nodo de arranque).

3.3. Bundle

Los bundle son colecciones de charms que unen aplicaciones. Permiten a los administradores implementar sistemas grandes y complejos desde un archivo, donde toda la configuración y las relaciones están predefinidas. Mientras que un charm representa una sola aplicación o servicio, un bundle representa un modelo juju completo.

3.4. Relaciones

La forma en que Juju permite que los servicios se comuniquen entre sí, y la forma en que se agrupa la topología de los servicios. El Charm define qué relaciones ofrece a otros servicios (proporciona), y qué tipo de relaciones puede hacer con otros servicios (requiere).

En muchos casos, el establecimiento de una relación resultará en una conexión TCP real que se está creando entre las unidades de servicio, pero eso no es necesariamente el caso. También se pueden establecer relaciones para informar a los servicios de los parámetros de configuración, solicitar información de monitoreo o cualquier otro detalle que el autor del Charm haya elegido poner a disposición.

3.5. Servicios juju

Juju opera en términos de servicios. Un servicio de juju es cualquier aplicación (o conjunto de aplicaciones) que se integra en el marco como un componente individual que generalmente debería unirse con otros componentes para realizar una tarea más compleja. Como ejemplo, WordPress podría implementarse como un servicio y, para realizar correctamente sus tareas, podría comunicarse con un servicio de base de datos y un servicio de equilibrio de carga.

4. ESCENARIO

A continuación vamos a ver como sería nuestro escenario y que necesitaremos para que podamos instalar Mattermost. La lista de pasos son las siguientes:

1. Lo primero que tenemos que tener es la cuenta de AWS con un grupo de usuario y sus claves para así poder conectarla con juju y desplegar posteriormente en AWS.
2. Lo siguiente es conectar juju con aws con una lista de comando mediante la terminal.
3. Una vez conectado juju con aws tendremos que desplegar Mattermost y los complementos que necesite, en este caso necesitaremos un postgresql.

5. CASO PRÁCTICO JUJU + AWS

5.1. Instalación de juju

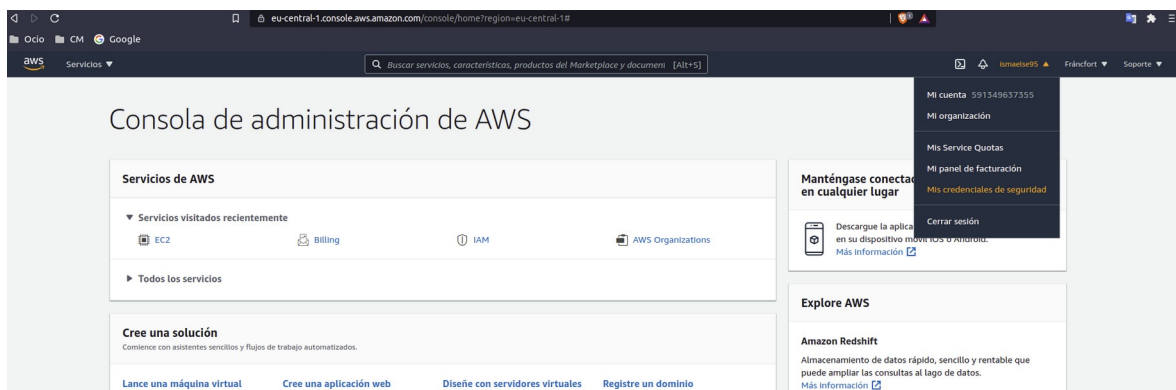
Lo primero que tendremos que hacer es instalar juju, para instalar juju en ubuntu tendremos que ejecutar el siguiente comando.

```
$ sudo snap install juju --classic
```

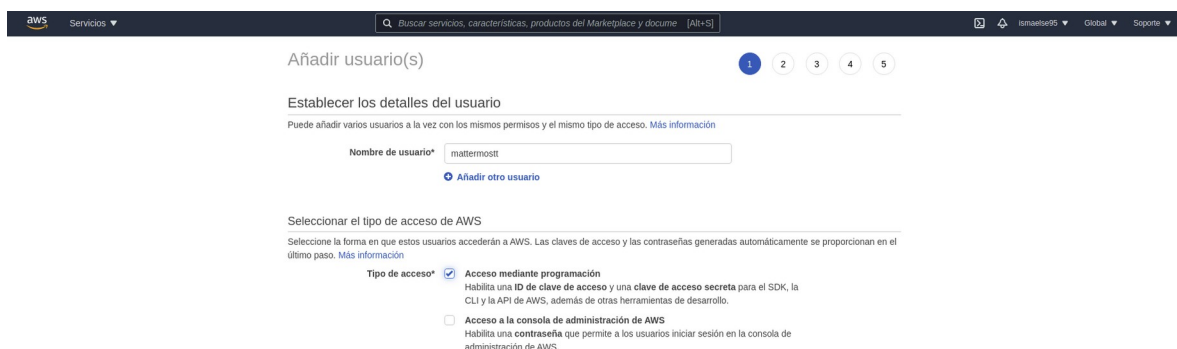
5.2. Creación de cuenta AWS

Vamos a continuar entrando en aws y vamos a crear un usuario y un grupo de usuario para poder obtener la cuenta con sus claves, para ello tendremos que hacer los siguientes pasos:

Le damos al nombre, y después a Mis credenciales de seguridad.



Una vez dentro le damos a usuario y crearemos un nuevo usuario. Le pondremos un nombre al usuario y marcamos la casilla “Acceso mediante programación”.



Añadimos un nuevo grupo con las políticas asociadas de “AdministratorAccess”.

Añadir usuario(s)

1 2 3 4 5

Establecer permisos

Añadir un usuario al grupo Copiar permisos de un usuario existente Asociar directamente las políticas existentes

Añada un usuario a un grupo existente o crea uno. El uso de grupos es una práctica recomendada para administrar los permisos de un usuario por funciones de trabajo. Más información

Añadir un usuario al grupo

Crear un grupo Actualizar

Buscar Mostrando 1 resultado

| Grupo | Políticas asociadas |
|--------------------|---------------------|
| ✓ mattermost-admin | AdministratorAccess |

Le damos a siguiente y se crea el usuario con nuestra clave de acceso y la clave de acceso secreta.

Añadir usuario(s)

1 2 3 4 5

Correcto

Ha creado correctamente los usuarios que se muestran a continuación. Puede ver y descargar las credenciales de seguridad de los usuarios. También puede enviar a los usuarios un correo electrónico con instrucciones para iniciar sesión en la consola de administración de AWS. Esta es la última vez que las credenciales estarán disponibles para descargarlas. Sin embargo, puede crear otras en cualquier momento.

Los usuarios con acceso a la consola de administración de AWS pueden iniciar sesión en:
<https://591349637355.signin.aws.amazon.com/console>

Descargar .csv

| Usuario | ID de clave de acceso | Clave de acceso secreta |
|--------------|-----------------------|-------------------------|
| ✓ mattermost | AKIAYTLZPMDV7MUKEMH3 | ***** Mostrar |

5.3. Conectar juju con aws

Para conectar juju con AWS nos vamos a nuestra terminal y vamos a añadir las credenciales a un fichero de configuración .yaml que usa juju para las conexiones, ese fichero lo encontramos en la siguiente ruta.

```
$ $HOME/.local/share/juju/credentials.yaml
```

Sabiendo esto ahora nos toca configurar ese fichero mediante juju, tendremos que seguir los siguientes pasos empezando por añadir las credenciales.

```
$ juju add-credential aws
```

Al iniciar este comando nos saldrá un asistente donde tendremos que especificar los siguientes pasos, introduciendo el nombre de la credencial y las autenticaciones.

```
$ juju add-credential aws
This operation can be applied to both a copy on this client and
to the one on a controller.
No current controller was detected and there are no registered
controllers on this client: either bootstrap one or register one.
Enter credential name: mattermost

Using auth-type "access-key".

Enter access-key: AKIAYTLZPMDV6HBJXGGP

Enter secret-key:

Credential "mattermost" added locally for cloud "aws".
```

Una vez añadida las credenciales tendremos que especificar que la región en la que trabajaremos dentro de AWS sea eu-west-1, para ello tendremos que especificarlo con el siguiente comando:

```
$ juju set-default-region aws eu-west-1
```

Con esto ya podríamos ver el fichero para ver si lo tenemos todo bien configurado antes de crear el controlador, vamos a verlo con el siguiente comando.

```
$ sudo cat $HOME/.local/share/juju/credentials.yaml
credentials:
  aws:
    default-region: eu-west-1
  mattermost:
    auth-type: access-key
    access-key: AKIAYTLZPMDV6HBJXGGP
    secret-key: *****
```

Ya comprobado el fichero vamos a crear el controlador para poder implementar Mattermost, bien dicho esto vamos a crear el controlador juju para aws para ello ejecutamos el siguiente comando.

```
$ juju bootstrap aws mattermost
Creating Juju controller "mattermost" on aws/eu-west-1
Looking for packaged Juju agent version 2.9.3 for amd64
Located Juju agent version 2.9.3-ubuntu-amd64 at https://juju-dist-aws.s3.amazonaws.com/agents/agent/2.9.3/juju-2.9.3-ubuntu-amd64.tgz
Launching controller instance(s) on aws/eu-west-1...
  - i-0483d1a6d673d6e90 (arch=amd64 mem=3.8G cores=1)-west-1a"
Installing Juju agent on bootstrap instance
Fetching Juju Dashboard 0.7.1
Waiting for address
Attempting to connect to 172.31.12.102:22
Attempting to connect to 54.246.242.112:22
Connected to 54.246.242.112
Running machine configuration script...
Bootstrap agent now started
Contacting Juju controller at 54.246.242.112 to verify
accessibility...

Bootstrap complete, controller "mattermost" is now available
Controller machines are in the "controller" model
Initial model "default" added
```

Una vez creado nuestro controlador en la nube ya podremos empezar a desplegar el charm de Mattermost. Para iniciar este charm tendremos que seguir los siguientes pasos.

Primero tendremos que hacer un deploy de mattermost y postgresql.

```
$ juju deploy cs:~tengu-team/mattermost
Located charm "mattermost" in charm-store, revision 5
Deployment under prior agreement to terms: isrg-lets-encrypt
Deploying "mattermost" from charm-store charm "mattermost",
revision 5 in channel stable
$ juju deploy postgresql
Located charm "postgresql" in charm-hub, revision 233
Deploying "postgresql" from charm-hub charm "postgresql", revision
233 in channel stable
```

Cunado tengamos lanzadas las dos máquinas tendremos que conectar la aplicación mattermost al postgresql para ello lo hacemos con el comando:

```
$ juju add-relation mattermost postgresql:db
$ juju expose mattermost
```

Con esta forma tan sencilla ya tendríamos nuestra aplicación funcionando, podemos ver una vez realizados estos comando como va el estado de la implementación con el siguiente comando.


```
$ watch -c juju status -color
```

```
Model Controller Cloud/Region Version SLA Timestamp
default mattermost aws/eu-west-1 2.9.3 unsupported 14:22:26+02:00

App Version Status Scale Charm Store Channel Rev OS Message
mattermost 5.2.1 active 1 mattermost charmstore stable 5 ubuntu Ready (http://63.32.96.230:8065 [Insecure! Please set fqdn!])
postgresql 12.7 active 1 postgresql charmhub stable 233 ubuntu Live master (12.7)

Unit Workload Agent Machine Public address Ports Message
mattermost/0* active idle 0 63.32.96.230 80/tcp,443/tcp,8065/tcp Ready (http://63.32.96.230:8065 [Insecure! Please set fqdn!])
postgresql/0* active idle 1 34.245.41.119 5432/tcp Live master (12.7)

Machine State DNS Inst id Series AZ Message
0 started 63.32.96.230 i-03315a8dbb2fe8580 xenial eu-west-1a running
1 started 34.245.41.119 i-02d19fddbba650c6c9 focal eu-west-1b running
```

5.4. Configuración e instalación charm Mattermost

Ahora vamos a pasar a configurar mattermost para que funcione correctamente para ello tendremos que introducir la configuración mostrada a continuación.

Mattermost
All team communication in one place, searchable and accessible anywhere

Let's create your account

Already have an account? [Click here to sign in.](#)

What's your email address?

Valid email required for sign-up

Choose your username

Username must begin with a letter, and contain between 3 to 22 lowercase characters made up of numbers, letters, and the symbols '.', '-' and '_'.

Choose your password

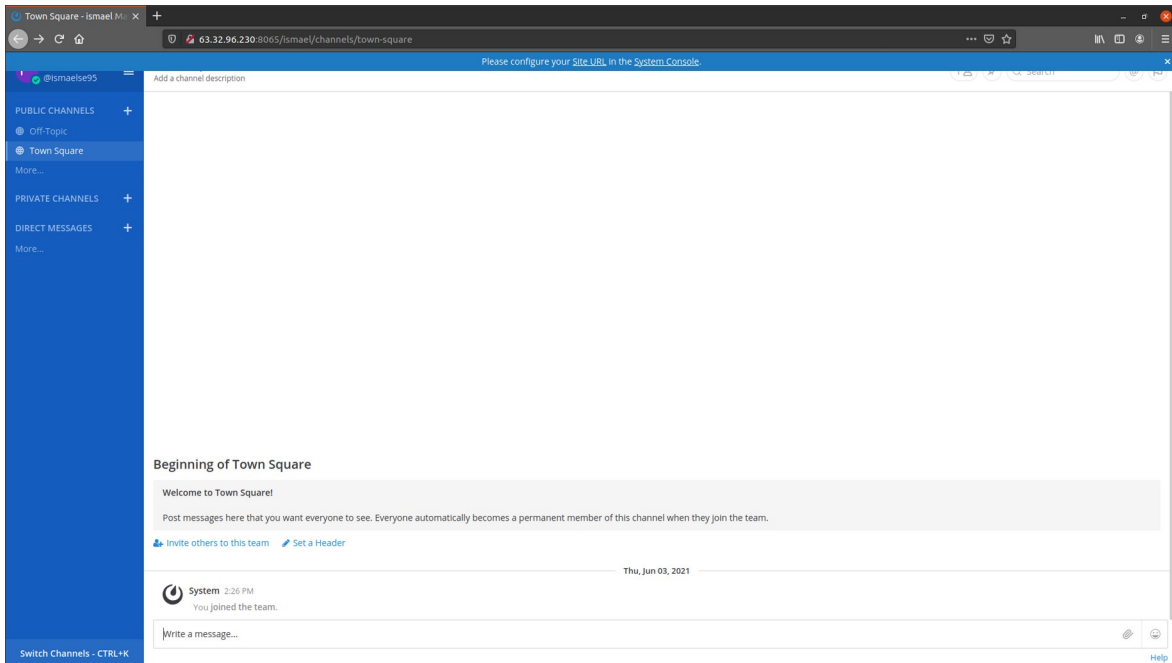
[Create Account](#)

By proceeding to create your account and use Mattermost, you agree to our [Terms of Service](#) and [Privacy Policy](#). If you do not agree, you cannot use Mattermost.

MATTERMOST

[About](#) [Privacy](#) [Terms](#) [Help](#) © 2015-2021 Mattermost, Inc.

Una vez introducidos los datos nos accederá directamente al panel para poder empezar a chatear.



5.5. Interfaz GUI

Vamos a probar también un charm que facilita el despliegue de una interfaz gráfica de usuario juju en un entorno existente.

Navegadores compatibles: La interfaz gráfica de usuario de juju admite las versiones recientes de Chrome, Chromium, Firefox, Safari y los exploradores web de Internet Explorer.

Servidores de demostración y puesta en escena: La interfaz gráfica de usuario de juju ejecuta un entorno de demostración en Demo.jujucharms.com. Desde allí se puede ver los Charms, probar la GUI y construir un entorno de ejemplo para exportar a fin de ser usado en otra parte.

Para empezar a utilizar con esta herramienta de juju tendremos que ejecutar un deploy de juju-gui para ello ejecutamos el siguiente comando.

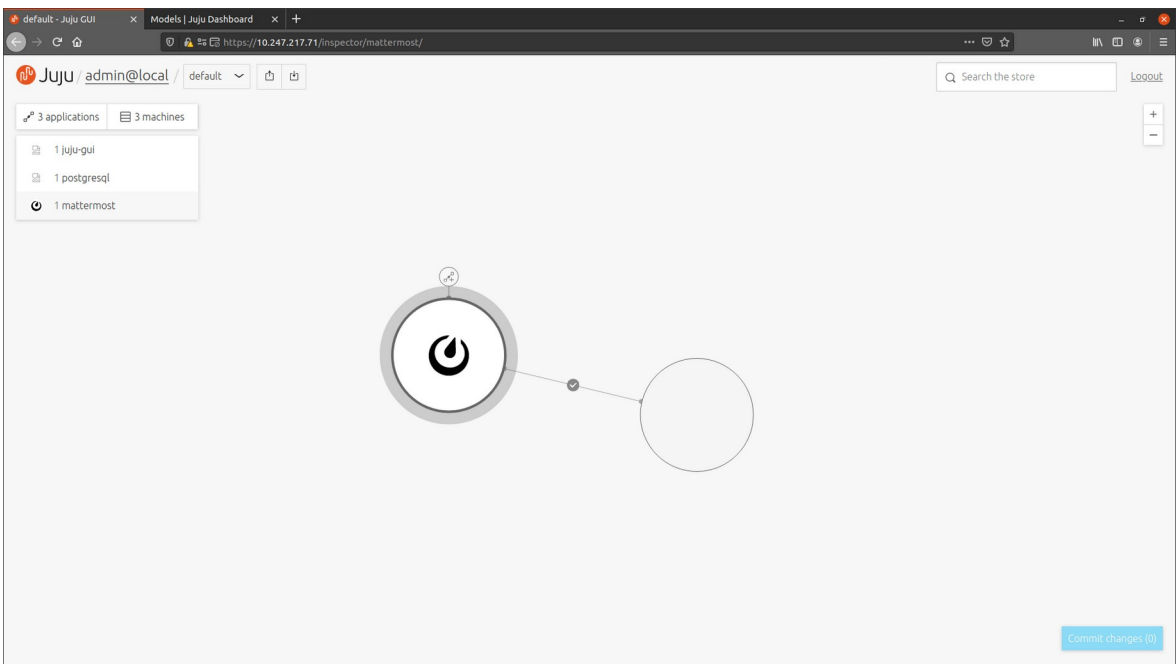
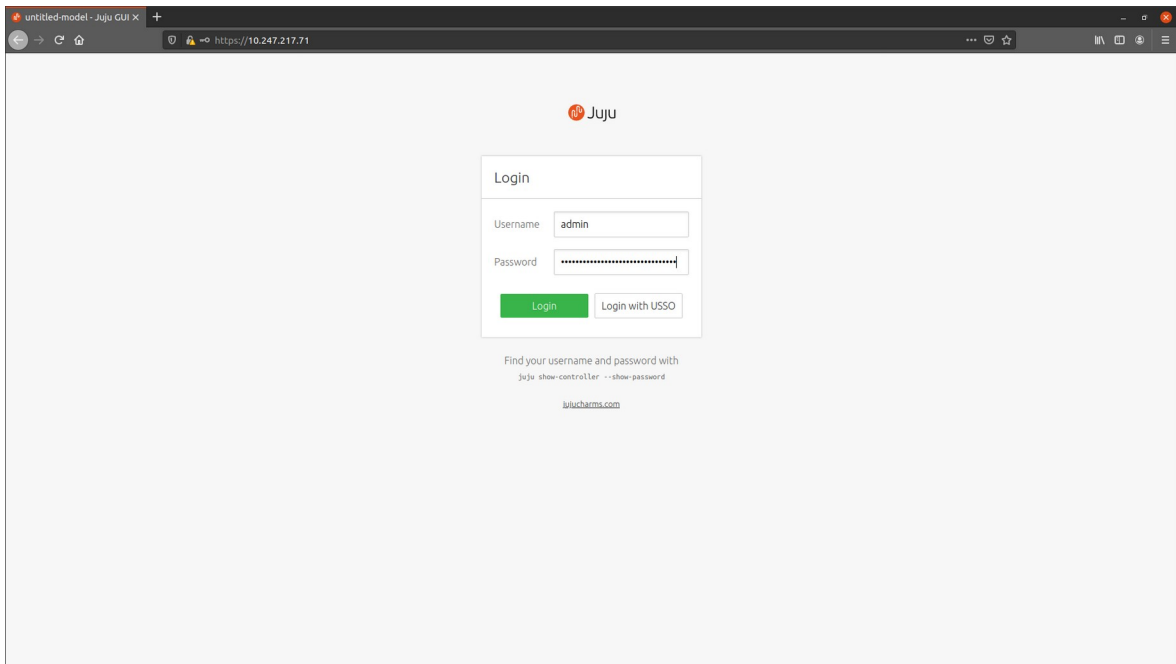
```
$ juju deploy juju-gui
```

Una vez desplegado vamos a obtener la contraseña que nos da por defecto. Ejecutamos el

siguiente comando.

```
$ juju show controller --show-password
```

Con esto accedemos a la página que también nos la da al pedir la contraseña y accedemos a ella para empezar a desplegar charms o ver los que ya tenemos creados.



5.6. Borrado de aplicación

Si queremos borrar nuestra aplicación, lo podremos hacer muy fácilmente desde nuestra terminal usando los siguiente comandos, primero borraremos la aplicación mattermost y la de postgresql.

```
$ juju remove-application mattermost
removing application mattermost
$ juju remove-application postgresql
removing application postgresql
```

5.7. Borrado de credenciales

Podremos borrar también nuestras credenciales creadas en este caso con AWS, primero podemos ver las credenciales que tenemos creadas con el comando:

```
$ juju credentials
No credentials from any controller to display.

Client Credentials:
Cloud Credentials
aws mattermost
```

En nuestro caso vamos a borrar mattermost así que usaremos el siguiente comando:

6. CONCLUSIÓN

En conclusión juju es una forma muy simple de desplegar aplicaciones como hemos visto, también a parte de AWS se puede conectar con nubes como Azure, google cloud y entre otras.

Ademas juju posee un amplio catalogo de charms con los que podemos ver muchas aplicaciones y desplegar cualquiera de ellas. También a parte de los charms juju cuenta con Bundles que son colecciones de los charms y unen múltiples aplicaciones con los cuales permiten a los administradores implementar sistemas grandes y complejos desde un archivo, donde toda la configuración y las relaciones están predefinidas.

7. BIBLIOGRAFÍA

1. [Página doc de juju](#)
2. [Página doc de AWS](#)