

# Laboratorio de Sistemas Operativos y Redes

# Mattermost

Ramiro Lescano - Juan C. Hualampa

10 de diciembre de 2015

## 1 Qué es Mattermost?

Mattermost es una alternativa a Slack de código abierto, este servicio organiza toda la comunicación de tal manera que la mensajería y el intercambio de archivos ocurren en un solo lugar y es accesible a través de computadoras y teléfonos celulares. Está desarrollado en Go y React, y su ejecución se realiza sobre una plataforma de Docker. Utiliza Ruby y la gema compass para compilar sass.

Hay varias formas de instalar la aplicación, vamos a mostrar los pasos para una instalación local.

## 2 Ventajas

Al ser self-hosted, brinda la posibilidad de comunicar grupos de trabajo, sin intercambiar información a través de servidores privados externos. Permite:

- Enviar mensajes, comentarios, imágenes y archivos a través de canales públicos o privados
- Acceso desde cualquier lugar, mediante pc o celular
- Importación de cuentas de Slack (beta, ver sección aparte)
- Integración mediante webhooks con otros servicios (gitHub, Jira, Google-Calendar)

## 3 Desventajas

Debido a su reciente lanzamiento, la documentación y la información en el foro de la aplicación a veces resulta insuficiente.

## 4 Requerimientos

### 4.1 SO

Es obligatorio para compilar Mattermost que el SO sea de 64-bit. Entre los sistemas operativos que soportan el server están:

- Ubuntu
- Debian
- CentOS
- RedHat Enterprise Linux
- Oracle Linux

### 4.2 Hardware

Es variable dependiendo de la cantidad de usuarios. Según estimaciones de grupos de 10-100 usuarios por equipo son necesarios:

- cpu: 2 cores para 250 usuarios
- ram: 2 gb para 50 usuarios
- disco: un estimado inicial para el servidor son 600-800 MB. Luego, es variable dependiendo del tráfico del grupo/usuario (bajo, medio, alto). Por ejemplo: un equipo de 30 personas con uso medio (5-25 MB/usuario/mes), con un factor de seguridad de 2x requerirá 300-1500 MB de espacio en disco por un año, mediante el cálculo:

$$qtyUsers * usagePerUser * safetyFactor$$

## 5 Instalación

### 5.1 Componentes requeridos

- Docker 64-bit
- Ruby
- Nodejs
- Go

## 5.2 instalar docker

```
sudo apt-get update
sudo apt-get install wget
wget -q0- https://get.docker.com/ | sh
sudo usermod -aG docker <nombre de usuario >
sudo service docker start
newgrp docker
```

## 5.3 setear host docker

```
sudo vim /etc/hosts (u otro editor de texto), insertar 127.0.0.1 dockerhost
newgrp docker
```

## 5.4 instalar build essentials

```
sudo apt-get install build-essential
```

## 5.5 instalar Go

descargar Go 1.5.1 de <http://golang.org/dl> , extraer en /usr/local

## 5.6 inicializar el workspace, agregar Go al path y editar el .bashrc

```
mkdir /go
sudo vim /.bashrc, insertar:
    export GOPATH=$HOME/go
    export GOROOT=/usr/local/go
    export PATH=$PATH:$GOROOT/bin
    ulimit -n 8096
```

## 5.7 recargar el bashrc para que tome los cambios

```
source /.bashrc
```

## 5.8 instalar node.js

```
curl -sL https://deb.nodesource.com/setup_5.x | sudo -E bash -
sudo apt-get install -y nodejs
```

## 5.9 instalar ruby y compass (para poder compilar el código sass)

```
sudo apt-get install ruby
sudo apt-get install ruby-dev
gem install compass
```

## 5.10 descargar mattermost

```
ir a la carpeta del workspace de go: cd /go
mkdir -p src/github.com/mattermost
cd src/github.com/mattermost
git clone https://github.com/mattermost/platform.git
cd platform
```

## 5.11 correr los tests unitarios para verificar la instalación

```
make test
```

## 5.12 una vez los tests hayan pasado, correr Mattermost

```
make run
```

# 6 Problemas

## 6.1 Intento de instalación en ubuntu 14.04 32-bits

Si bien la documentación de instalación de la aplicación dice que debe hacerse sobre un entorno Docker de 64 bits, intentamos realizar la compilación sobre un entorno de 32 bits, siguiendo los pasos de este [blog](#). Una vez instalado, obtenemos un error en la compilación de Mattermost.

## 6.2 Instalación paso a paso

Tuvimos problemas al instalar mattermost paso a paso siguiendo la guía de instalación, inconveniente que pudimos solucionar ejecutando la siguiente línea:  
`docker run --name mattermost-dev -d --publish 8065:80 mattermost/platform`  
parámetros:

- name: nombre del container
- -d: detached mode (corre sin necesidad de mantener abierta la terminal desde que se corrió)

- `wget -q0- https://get.docker.com/ | sh`
- `-publish`: declara el puerto en el que se va a exponer el servicio del contenedor.
- `mattermost/platform`: el usuario `mattermost` de `docker` creo un build automatico que se baja de <https://github.com/mattermost/platform> .

De esta forma se instala y corre el contenedor de `docker`, así como también la imagen de `mattermost`, pero desconociendo la ubicación de los archivos de configuración y los contenedores. Además, una vez creado el container mediante el comando no puede volver a ejecutarse (crearse) bajo el mismo nombre.

## 7 Ubicación de los archivos de configuración, contenedores y logs

- `containers`:

```
/var/lib/docker/containers
```

- `config.json` de `mattermost`:

```
/var/lib/docker/aufs/mnt/{nombre del contenedor}/mattermost/config/config.json
```

- `log` de `mattermost`:

```
var/lib/docker/aufs/mnt/{nombre del contenedor}/mattermost/logs/mattermost.log
```

- base de datos `mysql`:

```
var/lib/docker/volumes/{nombre del contenedor}/_data/mysql/
```

- archivos compartidos en la plataforma:

```
/var/lib/docker/aufs/mnt/{nombre del contenedor}/mattermost/data/teams/{id del equipo}/channels
```

## 8 Opciones del archivo de configuracion `/config.json`

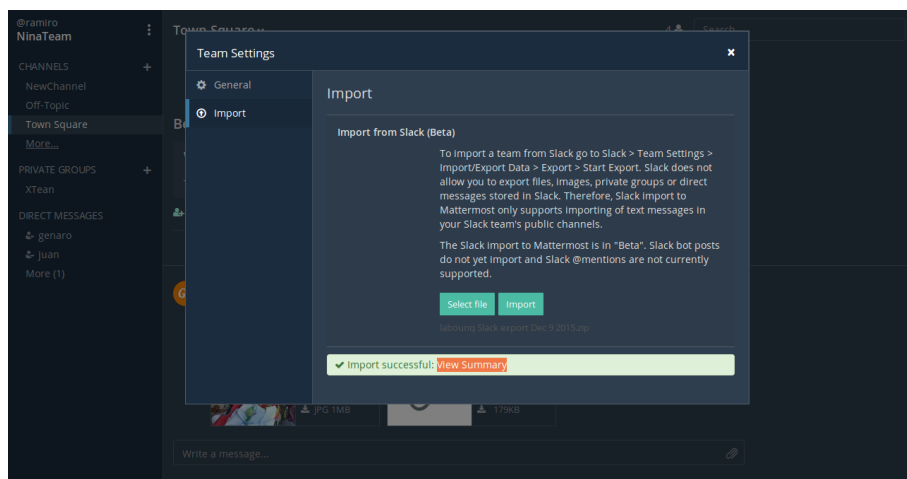
`Mattermost` puede configurarse desde línea de comando (ubicación detallada en punto 6) o desde la interfaz gráfica. Algunas de las opciones son:

- Sistema
  - `"ListenAddress": ":8065"` Puerto asignado a la aplicación
  - `"MaximumLoginAttempts": 10` Máxima cantidad de intentos de logueo sin pedir reset de password
  - `"EnableTesting": false` Seteado en true carga cuentas de test para desarrollo

- Webhooks
  - "EnableIncomingWebhooks" / "EnableOutgoingWebhooks": true  
Habilita la opción de integrar Mattermost con otras plataformas
- Equipos
  - "MaxUsersPerTeam": 10 Cantidad de usuarios máxima por grupo
- Configuraciones SQL
  - "DriverName": "mysql" Setea el tipo de base de datos a utilizar
- Configuraciones de archivos
  - "DriverName": "local" Especifica que se usa el sistema de archivos local
  - "Directory": "./data/" Path a la carpeta de archivos
- Configuraciones de log
  - "EnableFile": true Se escribe el log en el path indicado por "FileLocation"
  - "FileLocation": "" Directorio a donde se escribe el path, por defecto es ./logs/mattermost/mattermost.log

## 9 Importación de información desde slack

Aún no es posible la importación de información desde Slack. Siguiendo los pasos de la aplicación:



Muestra un mensaje de **import successful**. Clickeando sobre View Summary se descarga un archivo que muestra lo siguiente:

```
MattermostImportSummary(2).txt (-~/Descargas) - gedit
Mattermost Slack Import Log

Users Created
=====

Unable to import user: juan

Channels Added
=====

Failed to import: general|
hola
Failed to import: random

Notes
=====

- Some posts may not have been imported because they were not
supported by this importer.
- Slack bot posts are currently not supported.

Texto plano  Anchura de la pestaña: 8  Ln 11, Col 26  INS
```

Reflejando el error en la importación de canales y usuarios. Esto es un known-issue de Mattermost.

## 10 Faltantes del tp

No pudimos exponer el servidor a internet, y por consiguiente no pudimos integrar la aplicación con otro servicio mediante webhooks (github, por ejemplo, según tutoriales de [Mattermost](#) o [GitHub](#)).

## 11 Fuera del alcance del tp

[Instalación en varios nodos](#)

## 12 Relevamiento de máquinas utilizadas

ubuntu 14.04 LTS 64 bit - RAM: 2gb - HDD: 20gb - procesador: i3 m370 2.4GHz - windows 7 64 bit - vmware 7.1.3

ubuntu 14.04 LTS 64 bits - RAM: 4gb - HDD: 175 gb - procesador: Intel CPU B950 2.10GHz x 2