Practicos Curso Documentation

OMniLeads

22 de abril de 2020

Laboratorios:

CAPÍTULO 1

Deploy de OMniLeads

En este laboratorio vamos a desplegar la Aplicación en sus diferentes posibilidades.

Pondremos énfasis en el despliegue ***Classic CentOS OML*** para estandarizar todo el trabajo de configuración que se realizará en los próximos laboratorios. De todas maneras también se cubren los deploys ***Docker prodenv*** y ***Docker Devenv***.

Con respecto al tipo de deploy, desde OMniLeads se alienta el uso del método *Deployer & Nodes*, lo cual permite gestionar múltiples instancias desde el repositorio clonado sobre la estación de trabajo del Admin.

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

1.1 Deploy Classic CentOS OML (AIO)

A partir de contar con una VM (10 GB de disco + 2 GB de RAM) con CentOS 7 (edición minimal) instalado y con acceso a internet SIN restricciones, llevar a cabo el deploy de la Aplicación, siguiendo el método (Self-Hosted ó Deployer & Nodes) que el alumno prefiera. Tener bien en cuenta que vamos a desplegar la rama «develop» de OMniLeads.

Importante:

```
git clone https://gitlab.com/omnileads/ominicontacto.git
cd ominicontacto
git checkout develop
```

Observar el **git checkout develop** que difiere del **git checkout master** indicado en la documentación.

Una vez finalizado el deploy, proceda a ingresar a su instancia utilizando la URL https://X.X.X.X (IP del CentOS Host). Luego seguir lo pasos indicados en la Documentación primer acceso a OMniLeads.

Importante: No olvide tomar un snapshot de la VM de manera tal que nos quede disponible una versión virgen de OMniLeads para en los últimos capítulos trabajar con *backup & restore*

1.2 Deploy Docker prodenv

A partir de contar con una VM (16 GB de disco + 2 GB de RAM) con Issabel-PBX instalado y con acceso a internet SIN restricciones, llevar a cabo el deploy de la Aplicación, siguiendo el método (Self-Hosted ó Deployer & Nodes) que el alumno prefiera.

Una vez finalizado el deploy, proceda a ingresar a su instancia utilizando la URL https://X.X.X.X:444 (IP del centos).Luego seguir lo pasos indicados en la Documentación primer acceso a OMniLeads.

1.3 Deploy Devenv

En este punto es requisito excluyente contar con una estación de trabajo basada en Debian 10, Ubuntu 18.04 (o posteriores). Si el alumno utiliza cualquier otro sistema operativo, puede ignorar este ítem de la práctica.

CAPÍTULO 2

Configuración Inicial

En este laboratorio vamos a configurar asuntos escenciales como:

- Agentes y Grupos de Agentes
- Supervisores
- Bitácora de audios personales
- Música de espera
- Pausas de Agentes
- Registro de la instancia

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

2.1 Crear usuarios y grupos

En este punto debemos crear:

- Un grupo de agentes (con las opciones de auto atención dialer y autoatención inbound activadas)
- Dos agentes dentro del mismo
- Un supervisor del tipo *supervisor administrador*.

2.2 Audios y música de espera

Subir el audio facilitado con el material del curso.

Crear una Playlist de MOH, utilizando como audio el archivo de MOH facilitado.

2.3 Pausas

Crear las pausas: coaching (productiva) y break (recreativa).

2.4 Registro de la instancia

Proceder con el registro de la instanacia de OMniLeads.

CAPÍTULO 3

Generación de banco de testing

Para poder ejecutar todos los laboratorios y/o testear de aquí hacia el futuro cualquier funcionalidad de la plataforma, dejamos en esta sección los pasos para implementar una instancia de Asterisk configurada como PSTN-Emulator o banco de pruebas.

Con los fines de generar el escenario *OMniLeads – SIP Trunk – ITSP (telco) – Subscribers*

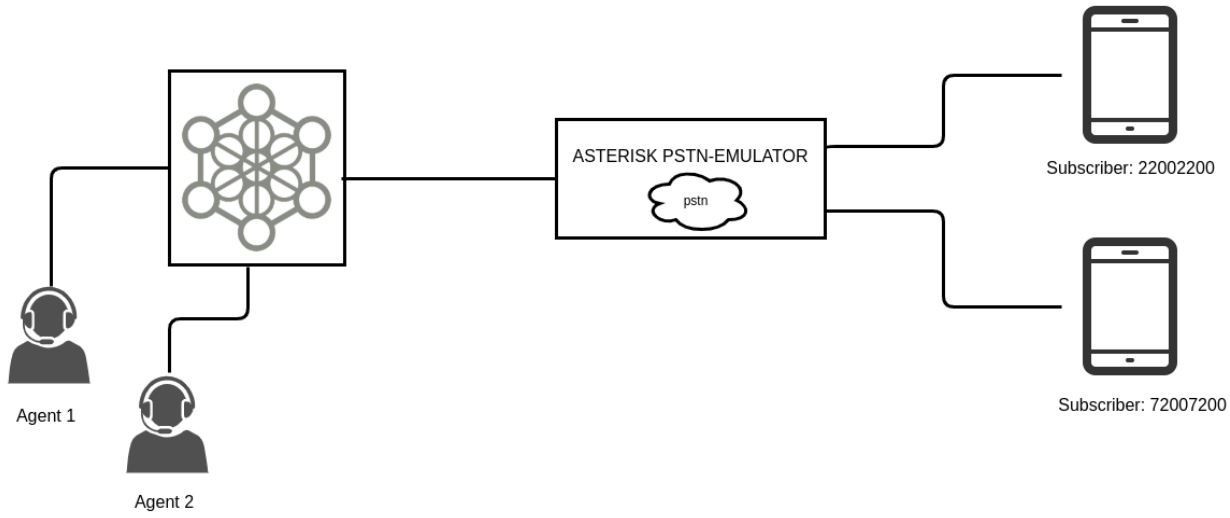


Figura: banco de pruebas

3.1 Generar VM con Asterisk

A partir de contar con una VM corriendo Ubuntu-Server o Debian, simplemente ejecutar el siguiente comando para instalar Asterisk.

```
sudo apt install asterisk
```

Nota: Es crucial que nuestra VM con Asterisk esté en el mismo segmento de red que la instancia de OMniLeads.

A partir de contar con Asterisk instalado vamos a trabajar con los archivos *sip.conf* y *extensions.conf*

Para el archivo sip.conf, copiar las siguientes lineas al final del mismo.

```
[subscribers](!)
type=friend
host=dynamic
dtmfmode=rfc2833
secret=omnileads
context=omnileads
qualify=yes

[72007200](subscribers)
[22002200](subscribers)

[40404040]
type=friend
host=dynamic
dtmfmode=rfc2833
secret=omnileads
context=omnileads
qualify=yes
```

Para el archivo extensions.conf se deberán añadir las siguientes lineas al final.

```
[omnileads]
exten => _[2,7][2,7]00[2,7][2,7]00,1,Verbose(2, from OML to test-phones)
same => n,Dial(SIP/${EXTEN})
same => n,Hangup()

exten => _4040404X,1,Verbose(2, from test-phones to OML)
same => n,Set(CALLERID(NUM)=${RAND(4140000,5140001)})
same => n,Dial(SIP/40404040/${EXTEN})
same => n,Hangup()

exten => _X.,1,Verbose(2, from OML to test)
same => n,Set(QUITAR=$[${LEN(${EXTEN})}-1])
same => n,GotoIf($["${EXTEN:${QUITAR}}" == "6"]?busy)
same => n,GotoIf($["${EXTEN:${QUITAR}}" == "7"]?completeOutnum)
same => n,GotoIf($["${EXTEN:${QUITAR}}" == "8"]?noAns)
same => n,GotoIf($["${EXTEN:${QUITAR}}" == "9"]?congestion)
same => n,Wait(5)
same => n,Answer()
same => n,Playback(demo-congrats)
same => n,Playback(demo-instruct)
same => n,Playback(tt-weasels)
same => n,Playback(demo-congrats)
same => n,Playback(demo-instruct)
same => n,Playback(tt-weasels)
same => n,Hangup()
same => n(busy),Busy(5)
same => n,Hangup()
```

(continué en la próxima página)

(proviene de la página anterior)

```
same => n(completeOutnum),Wait(5)
same => n,Answer()
same => n,Playback(tt-weasels)
same => n,Playback(tt-weasels)
same => n,Hangup()
same => n(noAns),Wait(120)
same => n,Hangup()
same => n(congestion),Congestion()
same => n,Hangup()
```

El dialplan aplicado se encarga de atender cualquier número marcado y reproducir una grabación, la idea es emular una llamada.

Números excepcionales:

- 40404040 al 40404049: si se marca cualquier número dentro de ese rango, se enviarán llamadas hacia OMni-Leads.
- 72007200: se envía la llamada a la cuenta 72007200.
- 22002200: se envía la llamada a la cuenta 22002200.
- Los números terminados en «6», darán tono de ocupado.
- Los números terminados en «7», serán atendidos, se pasan dos grabaciones cortas y se termina la llamada.
- Los números terminados en «6», darán tono de ocupado.
- Los números terminados en «8», no atenderán su llamada.
- Los números terminados en «9», darán congestion.

Bajo la premisa de emular la PSTN, también se buscó reproducir llamadas ocupadas, que no contesten o den congestion.

Finalmente ejecutar un *module reload* sobre el CLI de Asterisk para que se implemente nuestra configuración.

3.2 Registrar los softphones de prueba

En este punto se procede con la comprobación de la configuración aplicada, para ello vamos a registrar como suscritores a las cuentas SIP recientemente creadas. Estas son: 72007200 y 22002200.

Para ello podemos utilizar cualquier Softphone que corra sobre su estación de trabajo o smartphone.

Probar una llamada a cualquier número y comprobar que el Asterisk de prueba atiende la misma y reproduce una grabación.

CAPÍTULO 4

Configuración de salida a la PSTN

En esta sección vamos a configurar un troncal SIP entre la instancia de OMniLeads y el banco de pruebas o *PSTN Emulator*.

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

4.1 Crear PJSIP Trunk en OMniLeads

En este punto se debe generar un nuevo SIP Trunk utilizando PJSIP como módulo subyacente. Para el caso de este práctico aplica la plantilla *Custom*, ya que OMniLeads y el host PSTN-Emulator se encuentran dentro del mismo segmento de red (por ello el parámetro *transport=trunk-transport*), pero a diferencia de un escenario de proveedor con «backbone privado», aquí vamos a aplicar un registro de OMniLeads hacia el PSTN-Emulator (por ello el parámetro *sends_registrations=yes*)

Por lo tanto vamos a ingresar el siguiente bloque de configuración:

```
type=wizard
transport=trunk-transport
accepts_registrations=no
accepts_auth=no
sends_registrations=yes
sends_auth=yes
endpoint/rtp_symmetric=yes
endpoint/force_rport=yes
endpoint/rewrite_contact=yes
endpoint/timers=yes
aor/qualify_frequency=60
endpoint/allow=alaw,ulaw
endpoint/dtmf_mode=rfc4733
endpoint/context=from-pstn
remote_hosts=XXX.XXX.XXX.XXX:5060
endpoint/from_user=40404040
outbound_auth/username=40404040
outbound_auth/password=omnileads
```

Donde XXX.XXX.XXX.XXX es la IP del pstn-emulator.

4.2 Comprobar disponibilidad del trunk y registro

Para comprobar si el Asterisk de OMniLeads llega a ver disponible el trunk del otro extremo se debe ejecutar el comando:

```
asterisk -rx 'pjsip show endpoints'
```

En caso positivo, se debería visualizar un status de *Avail*.

```
centos*CLI> pjsip show endpoints

 Endpoint:  <Endpoint/CID.....................................>  <State.....>  <Channels.>
    I/OAuth:  <AuthId/UserName.....................................................>
        Aor:  <Aor.............................................>  <MaxContact>
      Contact:  <Aor/ContactUri..........................> <Hash....> <Status> <RTT(ms)..>
  Transport:  <TransportId........>  <Type>  <cos>  <tos>  <BindAddress..................>
   Identify:  <Identify/Endpoint.....................................................>
        Match:  <criteria.........................>
    Channel:  <ChannelId.....................................>  <State.....>  <Time.....>
        Exten: <DialedExten...........>  CLCID: <ConnectedLineCID.......>
==========================================================================================

 Endpoint:  1002/1002                                          Unavailable   0 of 1
        Aor:  1002                                           2
  Transport:  agent-transport           udp      0      0  0.0.0.0:5160

 Endpoint:  1003/1003                                          Unavailable   0 of 1
        Aor:  1003                                           2
  Transport:  agent-transport           udp      0      0  0.0.0.0:5160

 Endpoint:  1004/1004                                          Unavailable   0 of 1
        Aor:  1004                                           2
  Transport:  agent-transport           udp      0      0  0.0.0.0:5160

 Endpoint:  1005/1005                                          Unavailable   0 of 1
        Aor:  1005                                           2
  Transport:  agent-transport           udp      0      0  0.0.0.0:5160

 Endpoint:  pstn_emulator                                      Not in use    0 of inf
    OutAuth:  pstn_emulator-oauth/40404040
        Aor:  pstn_emulator                                  0
      Contact:  pstn_emulator/sip:192.168.43.230:5060      3f147571ff Avail         3.187
  Transport:  trunk-transport           udp      0      0  0.0.0.0:5161
   Identify:  pstn_emulator-identify/pstn_emulator
        Match: 192.168.43.230/32


Objects found: 5
```

Para comprobar si el Asterisk de OMniLeads se encuentra correctamente registrado en el proveedor SIP, lanzar el comando:

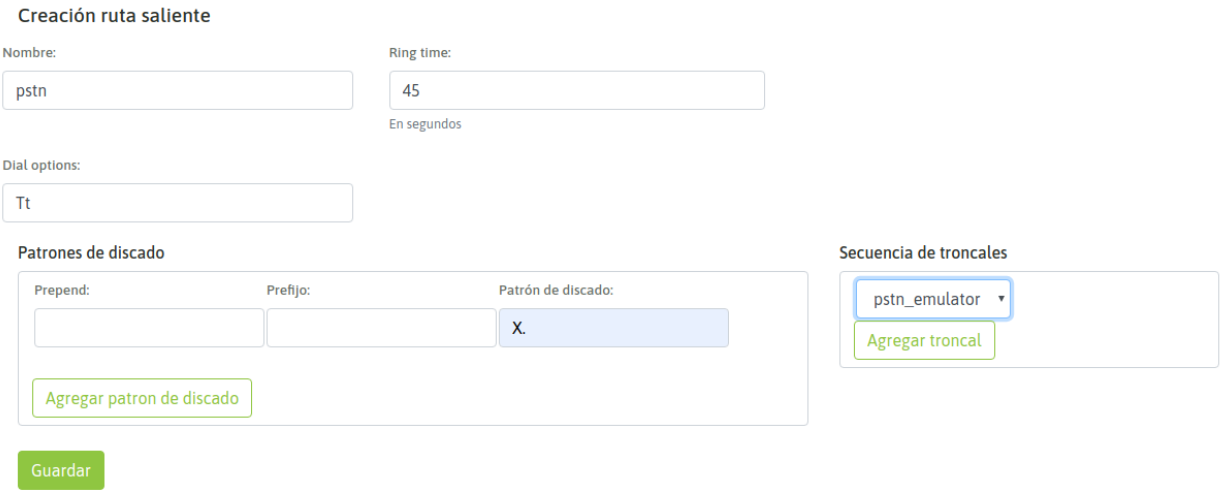
```
asterisk -rx 'pjsip show registrations'
```

En caso positivo, se debería visualizar un status de *Registered*.

```
centos*CLI> pjsip show registrations

 <Registration/ServerURI..............................>  <Auth..........>  <Status.......>
==========================================================================================

 pstn_emulator-reg-0/sip:192.168.43.230:5060            pstn_emulator-oauth  Registered

Objects found: 1
```

4.3 Generar ruta saliente

Finalmente se procede con la generación de una ruta saliente cuyo troncal de salida sea el trunk creado recientemente.



CAPÍTULO 5

Configuraciones comunes de campañas

En esta práctica vamos a generar alguno de los elementos comunes que toda campaña precisa relacionar para poder existir operativamente.

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

5.1 Bases de contactos

Subir al sistema las dos bases de contactos facilitadas.

5.2 Formularios

Generar dos formularios de campaña.

El primer formulario deberá implementar una encuesta similar a la siguiente:

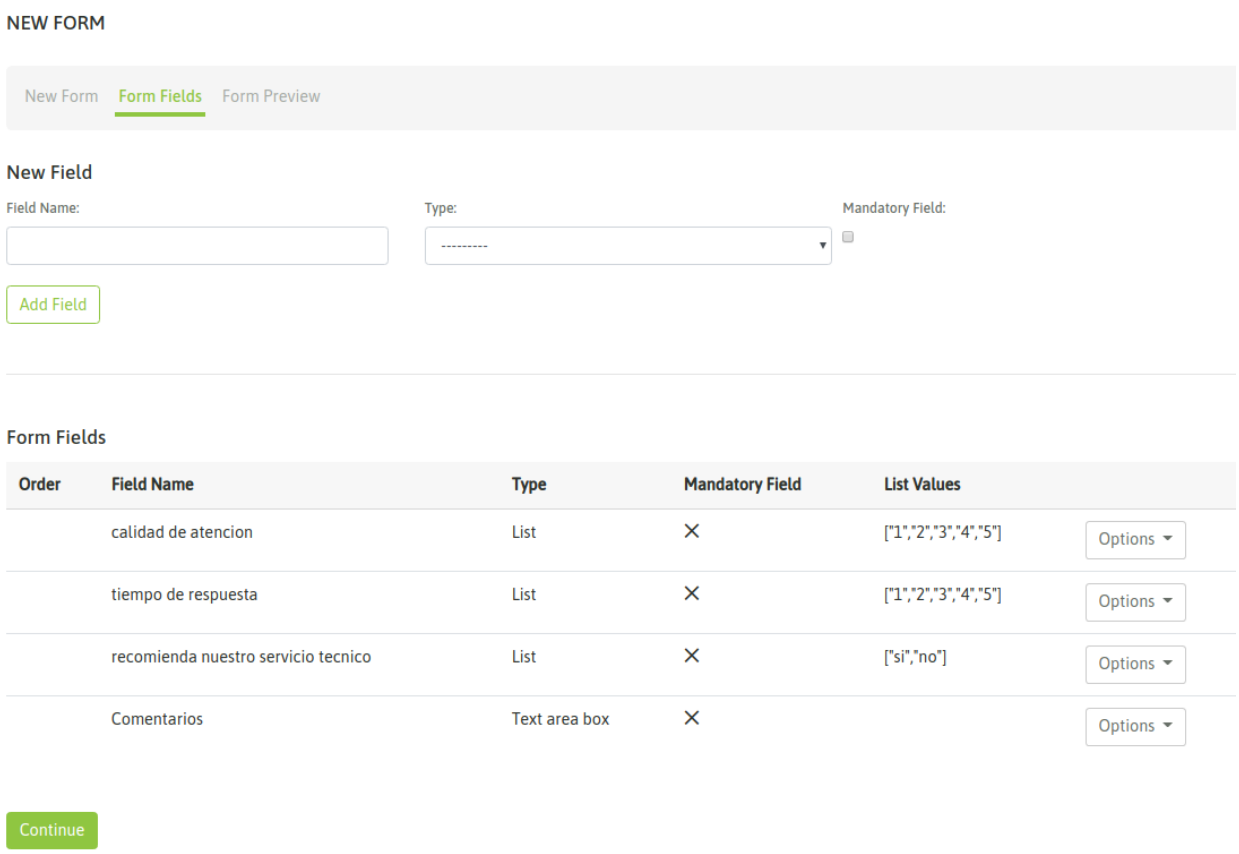


Figura: formulario de campaña A

El segundo formulario «venta» tendrá el siguiente aspecto:

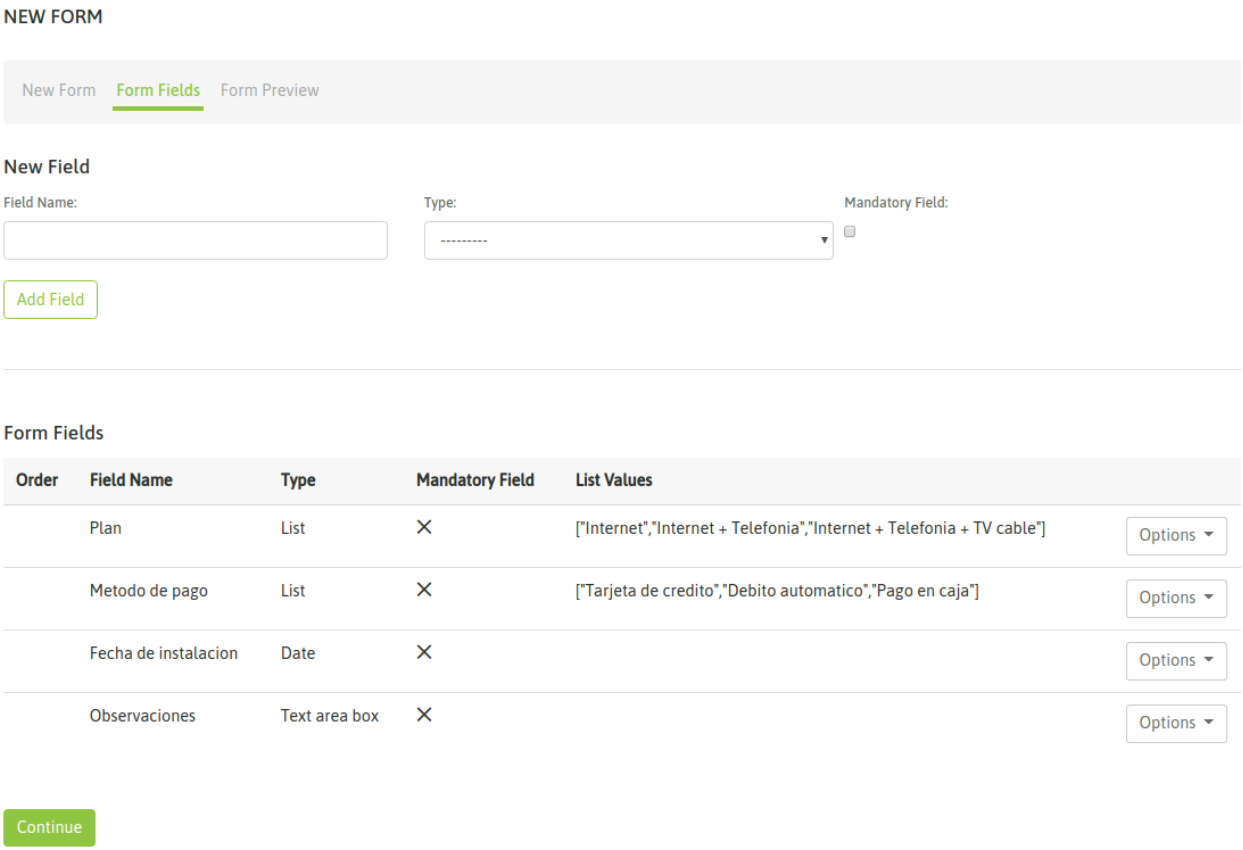


Figura: formulario de campaña B

5.3 Calificaciones

Generar las sigueintes calificaciones de agente:

- Ocupado
- No contesta
- Telefono erroneo
- Abonado fuera de servicio
- Equivocado
- Turno asignado
- Encuesta realizada

CAPÍTULO 6

Campañas preview

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

- Crear una campaña preview utilizando la base de contactos facilitada, el formulario «encuesta» creado y el lote de calificaciones.

La calificación «Encuesta realizada» deberá desplegar el formulario en cuestión.

- Ejecutar una serie de llamadas, probar diferentes calificaciones.
- Probar desactivar las grabaciones de la campaña y utilizar la grabación bajo demanda.

CAPÍTULO 7

Campañas manuales

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

Crear una campaña manual

No hace falta asociar una base de contactos, podemos asignar el formulario «encuesta» y el lote de calificaciones. La calificación «Encuesta realizada» deberá desplegar el formulario en cuestión.

Ejecutar una serie de llamadas y probar diferentes calificaciones

Marque algunas llamadas a números arbitrarios utilizando como salida la campaña manual generada en el paso anterior.

Luego puede probar generar una llamada manual a un número, guardalos como contacto y calificar el mismo.

CAPÍTULO 8

Campañas con discador predictivo

En esta sección se pretende poner en marcha las campañas con discador predictivo. Como es sabido, OMniLeads utiliza Wombat Dialer como componente crucial a la hora de generar las llamadas.

Por lo tanto debemos proceder con la configuración necesaria sobre dicho componente.

Luego se avanza con la configuración de campañas predictivas, utilizando la base de contactos facilitada para que el Dialer efectúe llamadas hacia nuestro PSTN-Emulator.

La siguiente sección de la Documentación Oficial cubre todos los aspectos relacionados a campañas de discado predictivo.

Crear una campaña

A continuación se pide crear una campaña del tipo predictiva utilizando la base de contactos más extensa y el formulario «Venta». La calificación «Venta» deberá disparar el formulario pertinente.

Ejecutar un reciclado

Una vez finalizado el recorrido de la base sobre la campaña, aplicar un reciclado de campaña.

Cambiar base de contactos de campaña

En este punto procederemos con un cambio de base de contactos sobre nuestra campaña existente.

Importante: Para llevar a cabo esta tarea es impresindible que el esqueleto de la base de contactos que desea asignar como reemplazo debe ser exactamente igual a la base reemplazada. Es decir independientemente de los contactos y la cantidad, las columnas deben ser exactamente iguales.

Aclarado este tema, procedemos con la aplicación de un cambio de base de contactos a nuestra campaña.

CAPÍTULO 9

Llamadas Entrantes

En este laboratorio vamos trabajar con el procesamiento de las llamadas entrantes. Al finalizar el capítulo deberíamos conseguir que nuestro OMniLeads implemente el siguiente esquema:

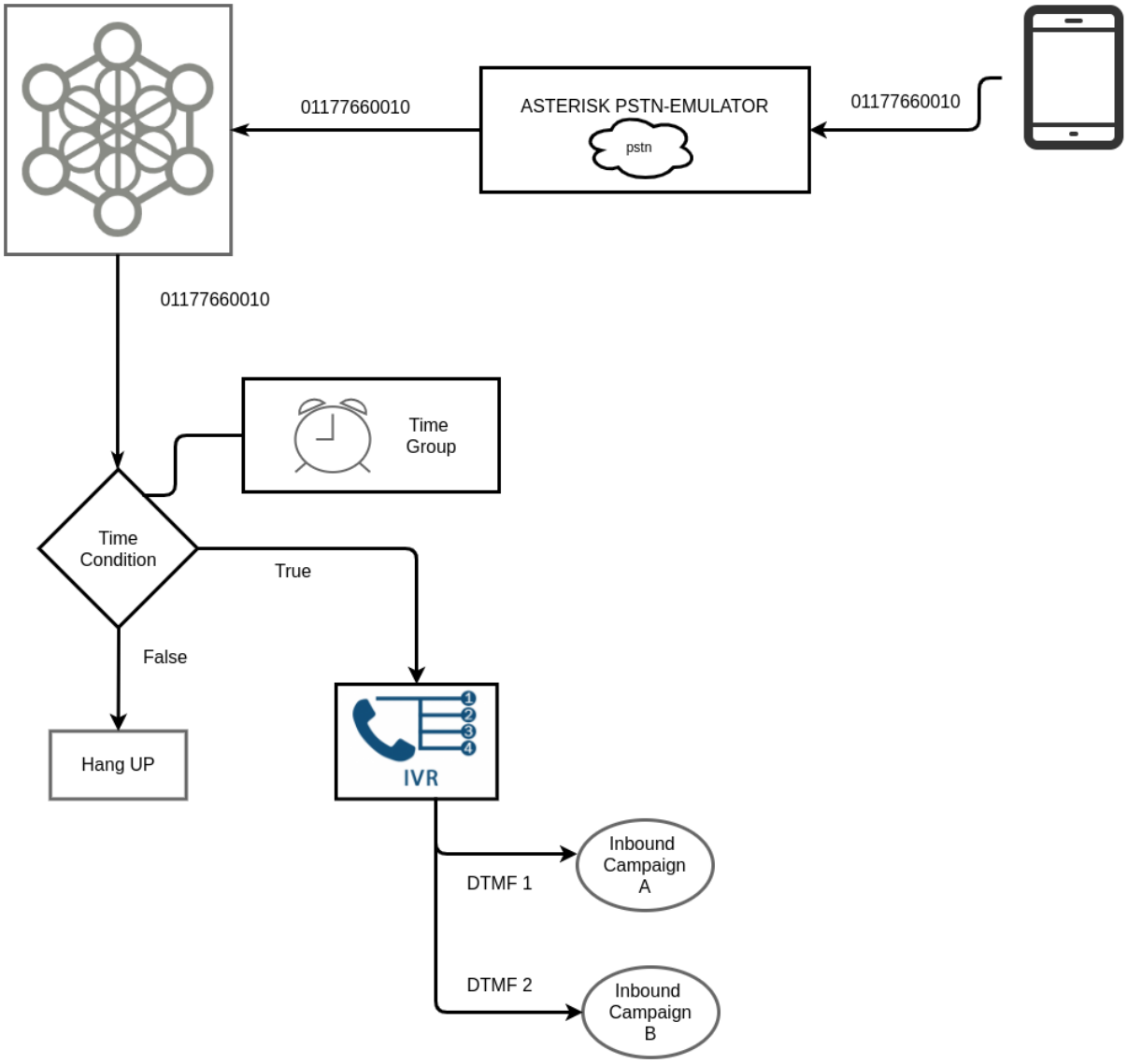


Figura: esquema implementado al finalizar este laboratorio

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

9.1 Creación de campañas entrantes

Crear dos campañas entrantes con un agente asignado a cada una. En las mismas podrá utilizar la MOH cargada durante anteriores laboratorios.

9.2 Enrutamiento

Generar una ruta entrante para el *DID: 40404040* cuyo destino sea nuestra cualquiera de las campañas entrantes.

Luego toma un softphone *subscriber* y ejecutar una llamada al número: *40404040*. Si la configuración es correcta, la llamada debería ingresar sobre la campaña asociada a la ruta entrante.

9.3 IVR

Crear un IVR utilizando nuestro *audio* facilitado. El menú deberá presentar para el *DTMF 1* que la llamada se encamine hacia la Campaña A, y para el *DTMF 2* hacia la Campaña B.

Generar una nueva ruta entrante para el *DID 40404041* y encaminar la llamada hacia el recientemente creado IVR. Para corroborar, tomar el softphone *subscriber* y llamar al número 40404041.

9.4 Validación condicionada por calendario

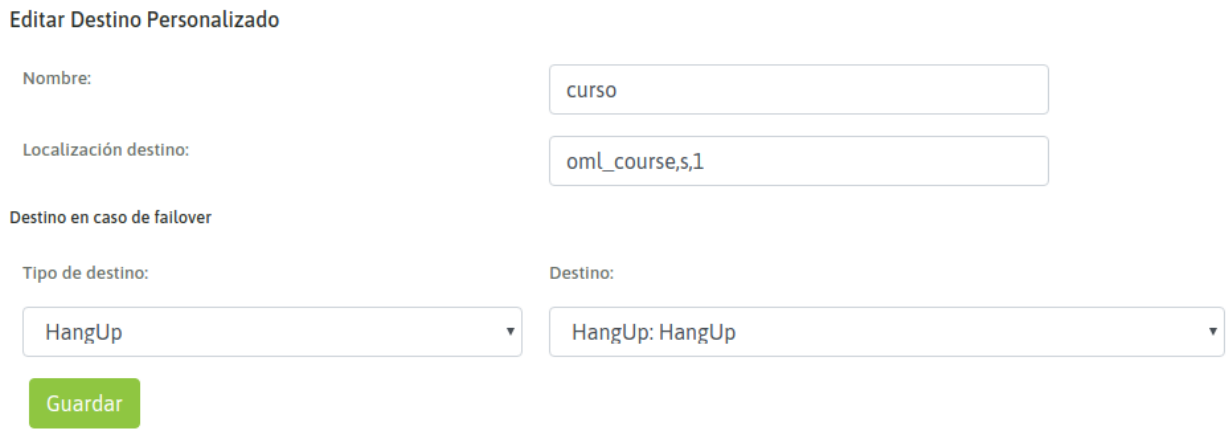
Generar un grupo horario *lun-vie 09:00 a 18:00*.

A continuación crear un condicional de tiempo asociado al grupo horario anterior, que en caso de coincidir dentro del rango horario del grupo la llamada se encamina hacia nuestro IVR mientras que en caso de caer fuera del rango horario del grupo, la llamada se envíe hacia un Hangup.

Generar una nueva ruta entrante para el *DID 40404042* y encaminar la llamada hacia el recientemente Condicional de Tiempo creado. Para corroborar, tomar el softphone *subscriber* y llamar al número 40404042.

9.5 Custom destinations

En este ítem vamos a generar un destino personalizado.

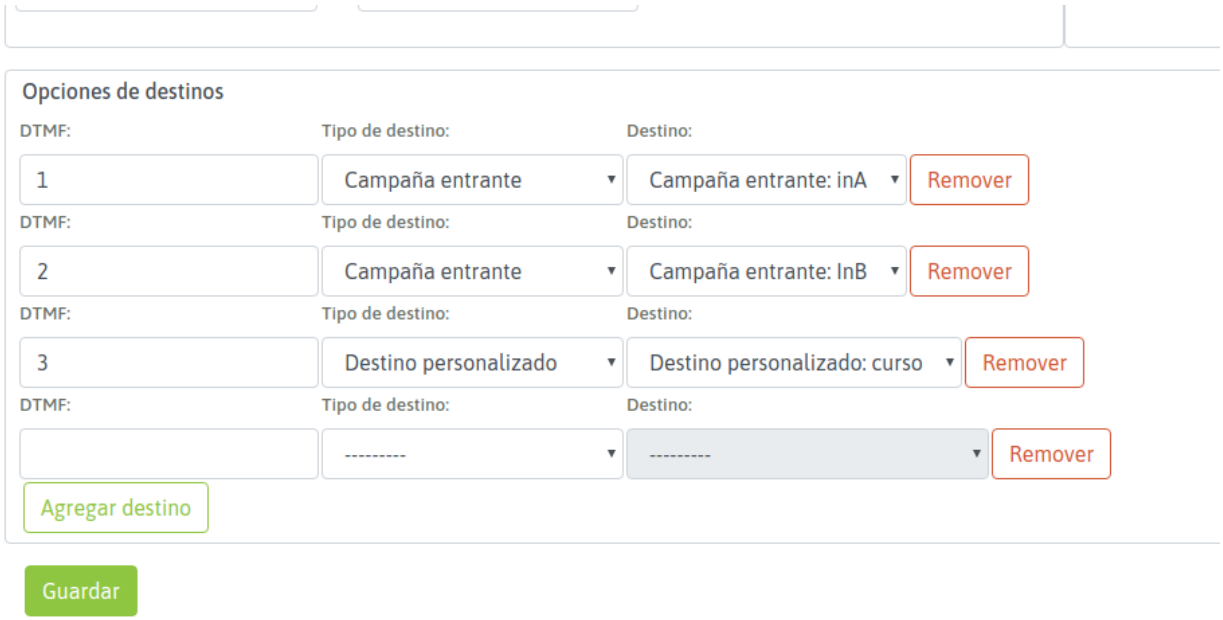


Una vez generado el *custom destionation* en OMniLeads, procedemos a generar el dialplan en el archivo pertinente para ello, es decir: **oml_extensions_custom.conf**

```
[oml_course]
exten => s,1,Verbose(2, custom dst oml course)
same => n,Playback(tt-weasels)
same => n,Hangup()
```

Luego hacer un «dialplan reload» en el CLI de Asterisk.

Finalmente, podemos añadir un nuevo *DTMF* dentro del IVR existente que invoque al destino personalizado.



Por último generar un llamado que caiga sobre el IVR e ingresar la opción que nos lleve al destino personalizado.

CAPÍTULO 10

Identificación de clientes en llamadas entrantes

Este módulo nos permite solicitar al llamante que ingrese su código de identificación, por lo tanto vamos a implementar dicha funcionalidad en nuestro diagrama de llamadas entrantes.

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

10.1 Identificación de llamada entrante e interacción con WS (servicio web) tipo 1

En este punto vamos a configurar una interacción con un servicio web al que simplemente pasamos el *ID de cliente* capturado por DTMF y esperamos del servicio que nos devuelva *true* o *false*. Si bien en este ambiente netamente pedagógico nosotros vamos a generar las respuestas utilizando la web la idea es utilizar esta funcionalidad para por ejemplo consultar al servicio web el estado de la cuenta del cliente que está llamando donde *true* puede significar que el cliente tiene su pago al día y *false* lo contrario, otra utilización es verificar si es un cliente ocasional o con un plan. En base a ello OMniLeads nos permite encaminar la llamada hacia un lugar u otro.

En primer lugar entonces generamos «el servicio web» utilizando la web mencionada. Vamos a generar 2 URLs una que devuelva *true* y otra *false*. Volviendo a recordar que esto es netamente una prueba de concepto y funcionalidad en OMniLeads.

URL que devuelve FALSE

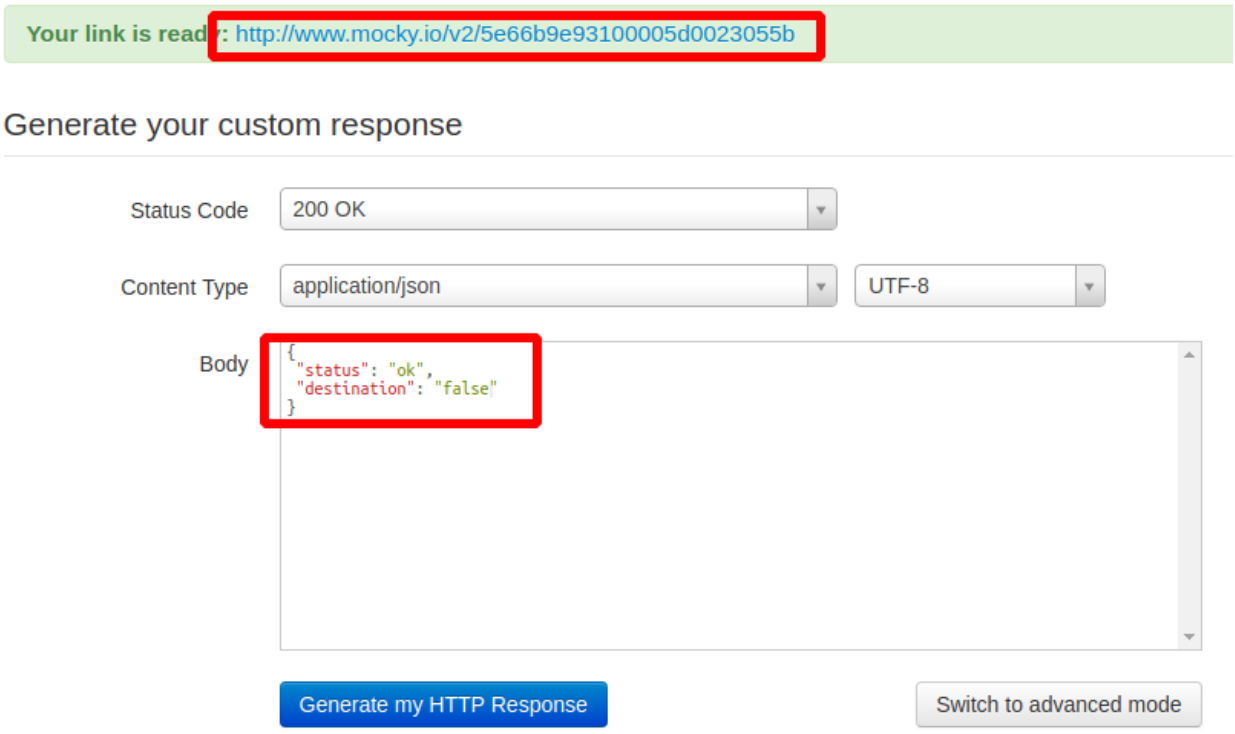


Figura: generacion de URL que devuelve FALSE

Al abrir el URL con el browser deberíamos observar la respuesta JSON generada.

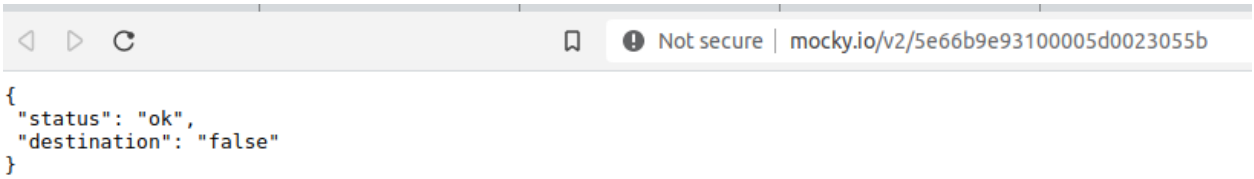


Figura: generacion de URL que devuelve FALSE

Por lo tanto GUARDAMOS esta URL-False porque la vamos a usar en los siguientes pasos.

URL que devuelve TRUE

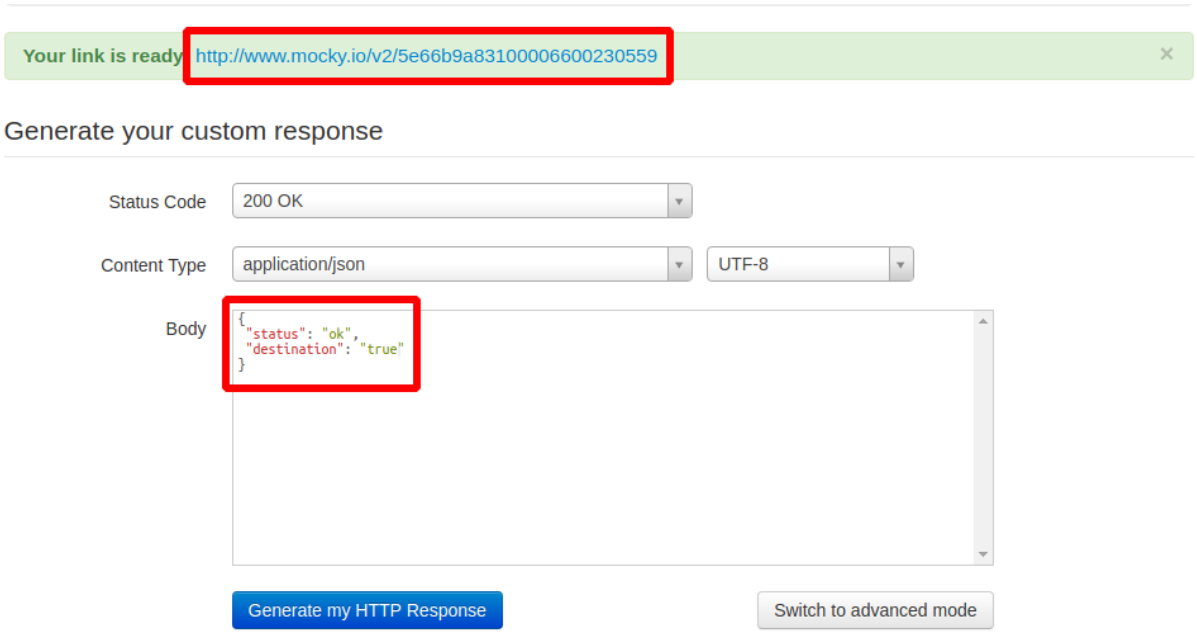


Figura: generacion de URL que devuelve TRUE

Al abrir el URL con el browser deberíamos observar la respuesta JSON generada.

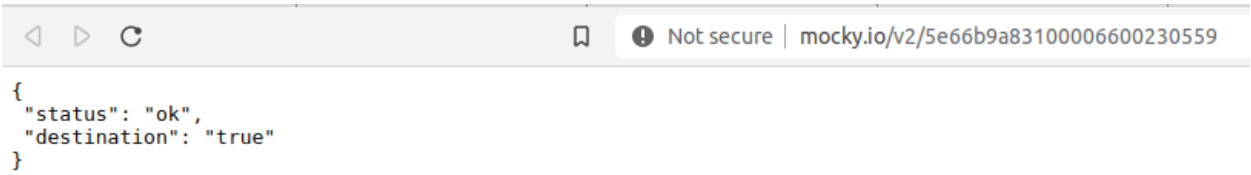


Figura: generacion de URL que devuelve TRUE

Por lo tanto GUARDAMOS esta URL-True porque la vamos a usar en los siguientes pasos.

A continuación debemos crear un objeto identificación de llamadas entrantes en OMniLeads:

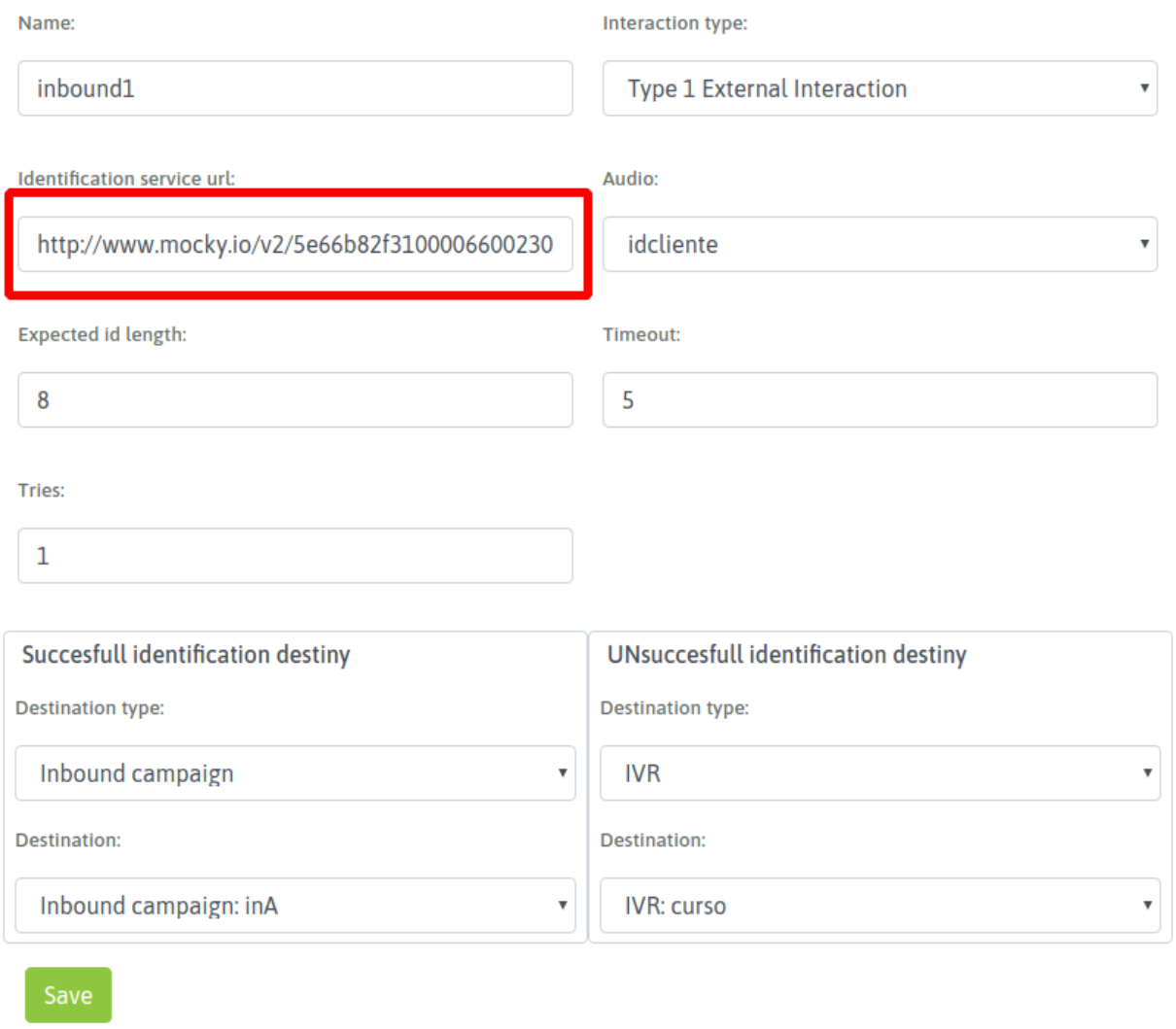


Figura: generacion del objeto identificación de llamadas

En la figura se remarca el campo en donde debemos insertar cada URL generada en los pasos anteriores. Podemos elegir hacia donde encaminar las llamadas si el «Servicio Web» devuelve *True* y hacia donde si es *False* la respuesta del JSON recibido.

Para poder derivar llamadas hacia la funcionalidad podemos generar una nueva ruta entrante:

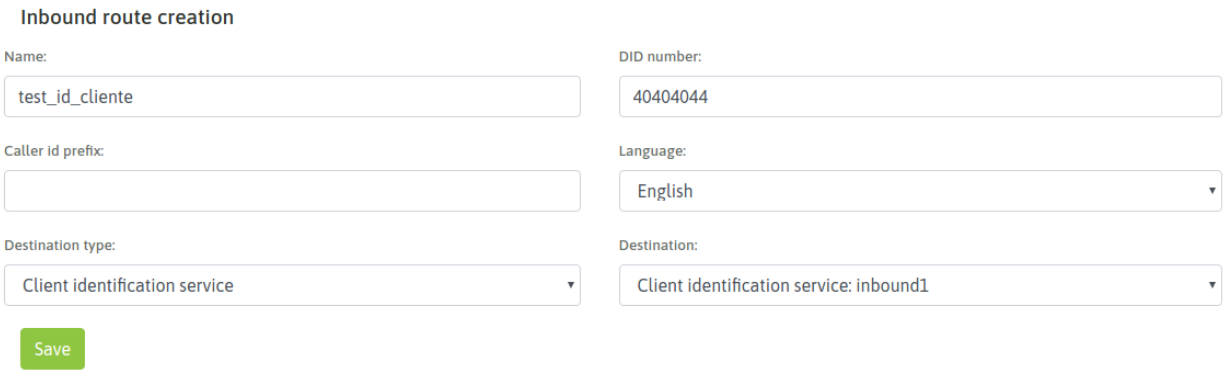


Figura: ruta entrante que deriva al objeto identificación de llamadas

Lo que finalmente nos queda por probar es:

- 1 - Utilizando el URL-False generar una llamada entrante al DID de nuestra ruta y comprobar que se envíe la llamada hacia el destino False.
- 2 - Utilizando el URL-True generar una llamada entrante al DID de nuestra ruta y comprobar que se envíe la llamada hacia el destino True.

Obviamente en un caso real no habrá que andar cambiando de URLs, simplemente una misma URL devolverá un JSON true o false basado en el ID que nosotros enviamos.

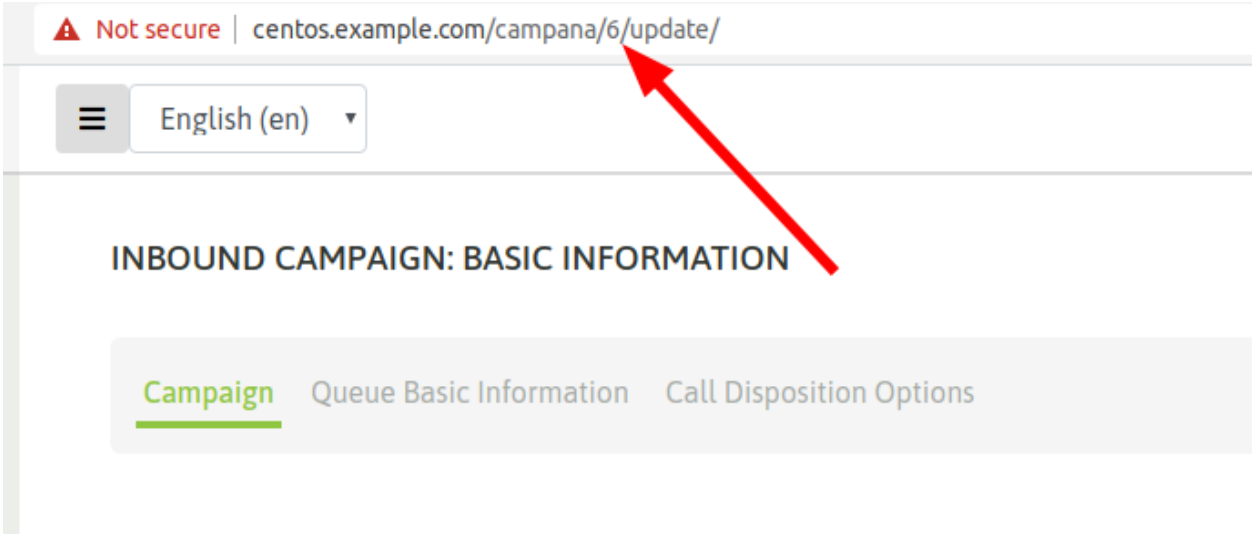
10.2 Identificación de llamada entrante e interacción con WS (servicio web) tipo 2

En este punto también consultaremos un servicio web, pero utilizando la configuración que permite que el servicio web nos devuelva una tupla (x,y) que indica hacia qué destino de OMniLeads hay que enviar la llamada.

OMniLeads identifica cualquier destino interno usando un par de valores (X = tipo de módulo, Y = objeto particular dentro del tipo de módulo). Por lo tanto el servicio web (por ejemplo un endpoint de la API de un CRM) podría directamente encaminar las llamadas hacia diferentes campañas entrantes. Una aplicación de esto podría ser encaminar las llamadas dependiendo de si el cliente que nos llamada tiene planes con mayor nivel de servicio, por lo tanto desde el CRM decidir sobre a que campaña enviar las llamadas de los clientes.

Como nosotros tenemos dos campañas entrantes generadas, vamos a utilizar nuevamente esta web para generar otros JSON funcionales a esta práctica.

Pero antes vamos a averiguar los *ID de campaña* de cada una de las campañas entrantes, para ello simplemente ingresar a ellas y revisar el URL en el browser.



Hacemos lo mismo con la otra campaña.

Ahora vamos a generar una URL similar a la siguiente figura:

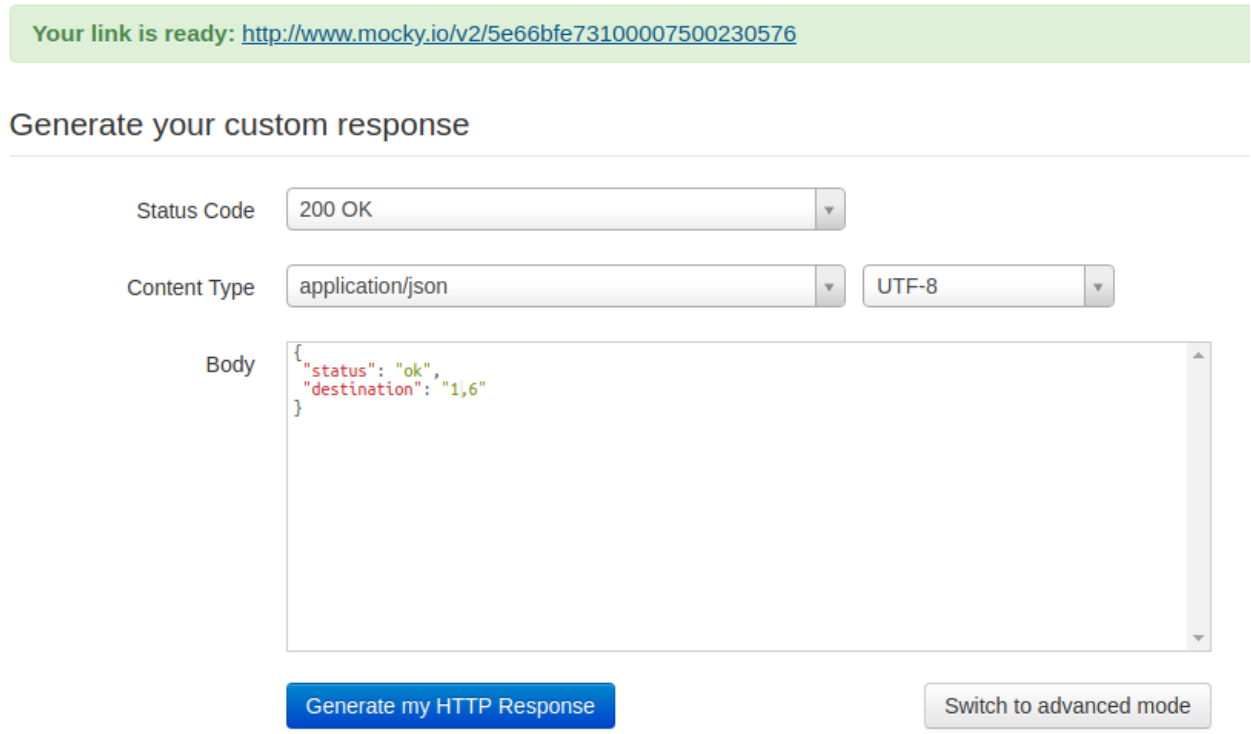
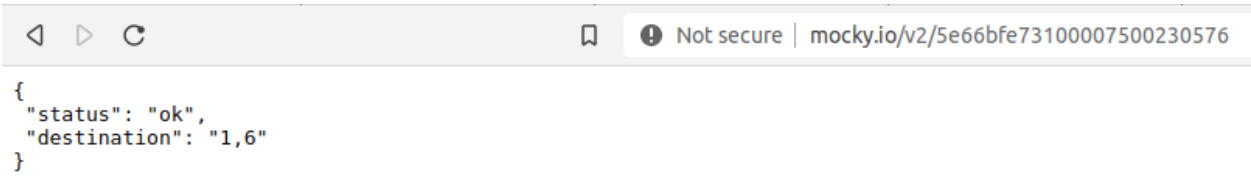
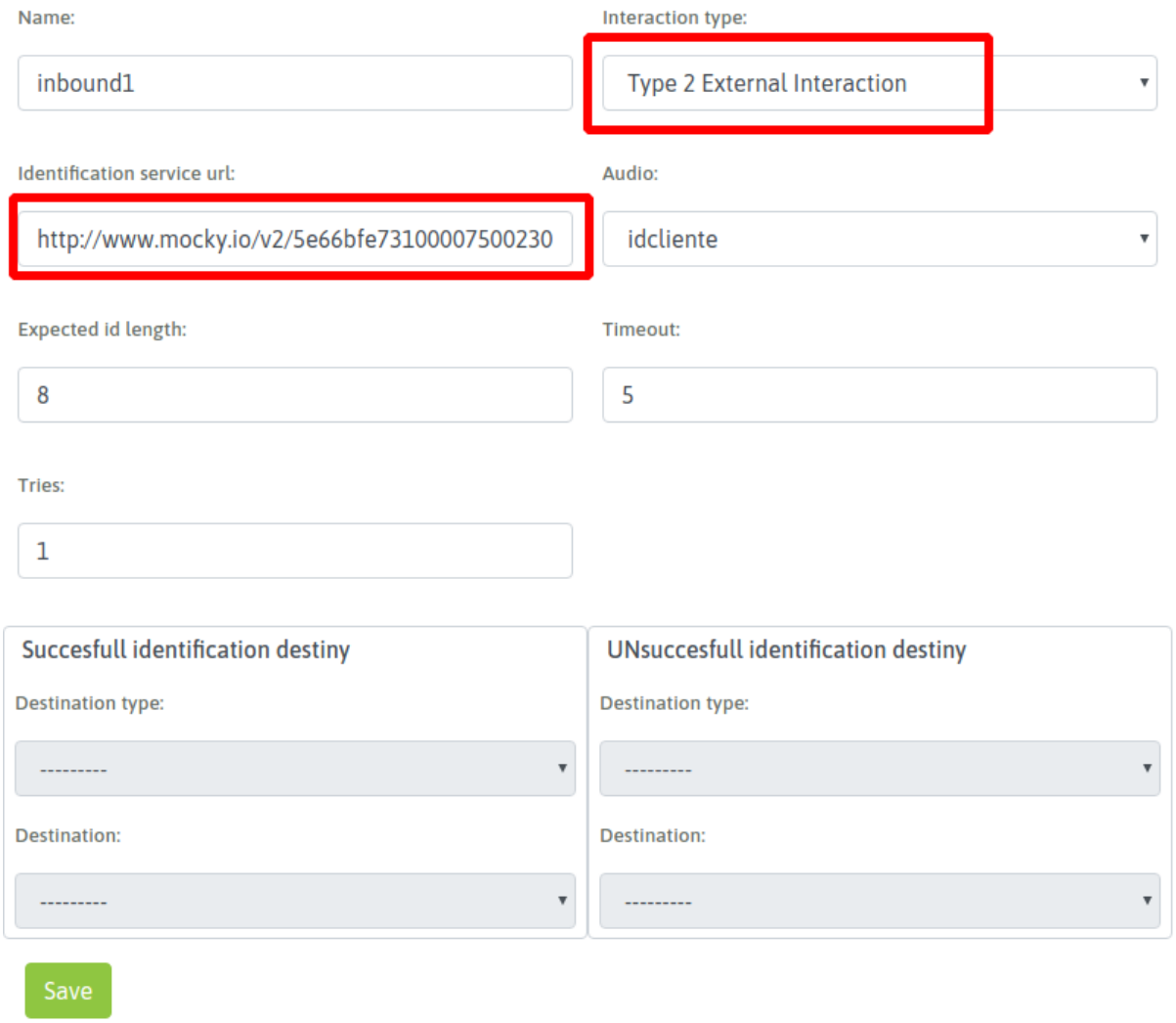


Figura: generacion de URL que devuelve el destino concreto a donde encaminar las llamadas



Como podemos ver la tupla que configuramos en el URL es (1,Y) donde 1=campaña entrante e *Y* es el id de la campaña en cuestión.

Ahora vamos a utilizar la URL recientemente generada para configurar un nuevo elemento *identificación de cliente* en OML como indica la siguiente figura:



Como muestra la figura, además de ingresar la nueva URL del servicio web, debemos modificar el campo *tipo de interacción* con el valor que implementa la interacción del tipo enrutamiento desde el servicio web.

Ahora podemos generar una nueva ruta entrante que apunte a este nuevo objeto y probar con una llamada. Si todo fue bien configurado entonces la llamada será enviada a la campaña cuyo ID se devuelve en el JSON.

CAPÍTULO 11

Integración con PBX

En este tramo del curso vamos a trabajar con la instancia *OMniLeads inside PBX*, que dejamos instalada en el práctico de deploys. Por lo tanto debemos encender la instancia y manos a la obra.

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

11.1 Troncalizar ambos sistemas

Troncal en OMniLeads

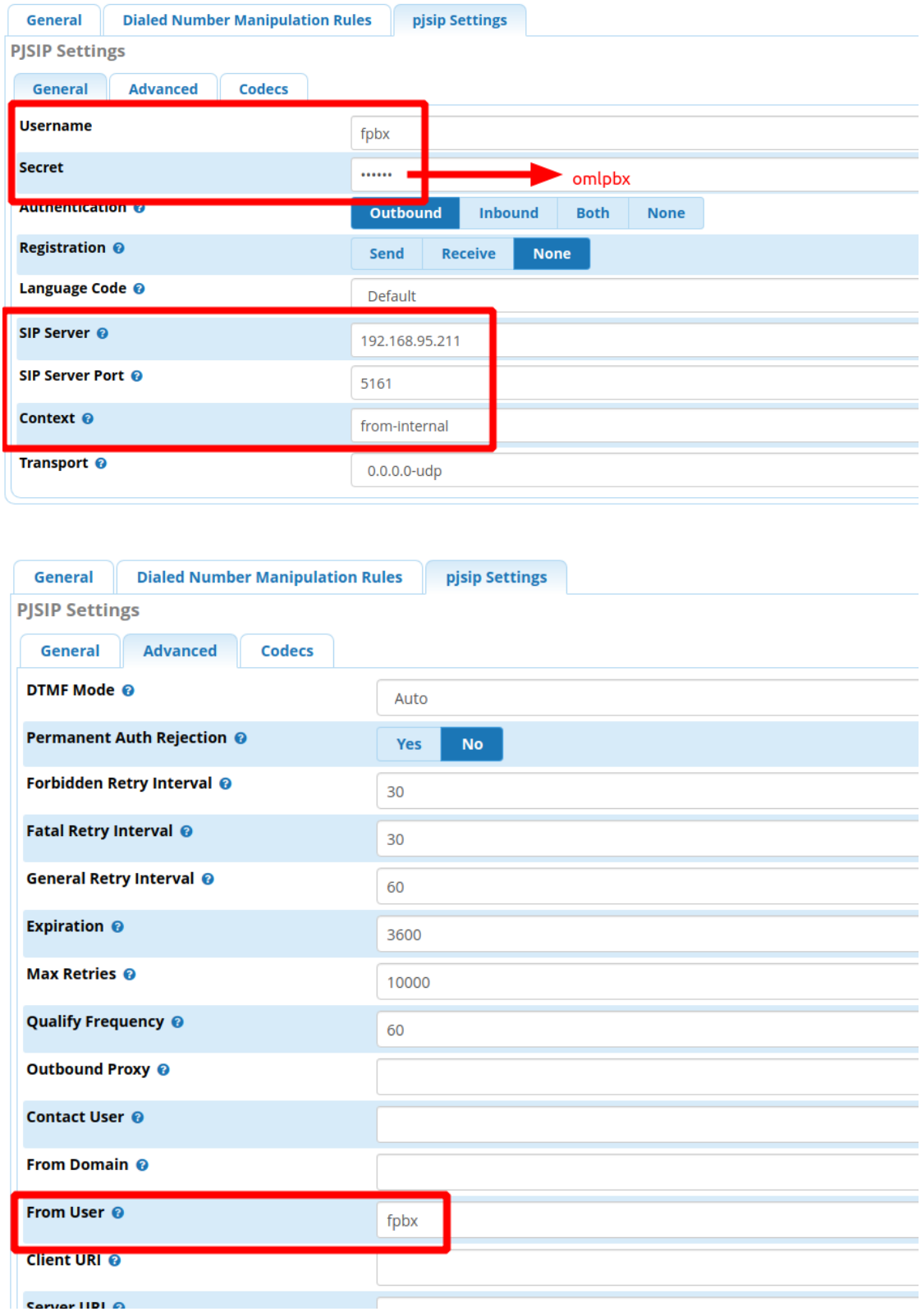
Utilizando la plantilla PBX LAN, vamos a considerar los últimos 6 parámetros:

```
inbound_auth/username=fpbx
remote_hosts=XXX.XXX.XXX.PBX:5060
inbound_auth/password=omlpbx
outbound_auth/username=omnileads
outbound_auth/password=omlpbx
endpoint/from_user=omnileads
```

Dónde **XXX.XXX.XXX.PBX** es la dirección IP de FreePBX

Troncal en FreePBX

Nos ajustamos a los valores que presentan la figura:

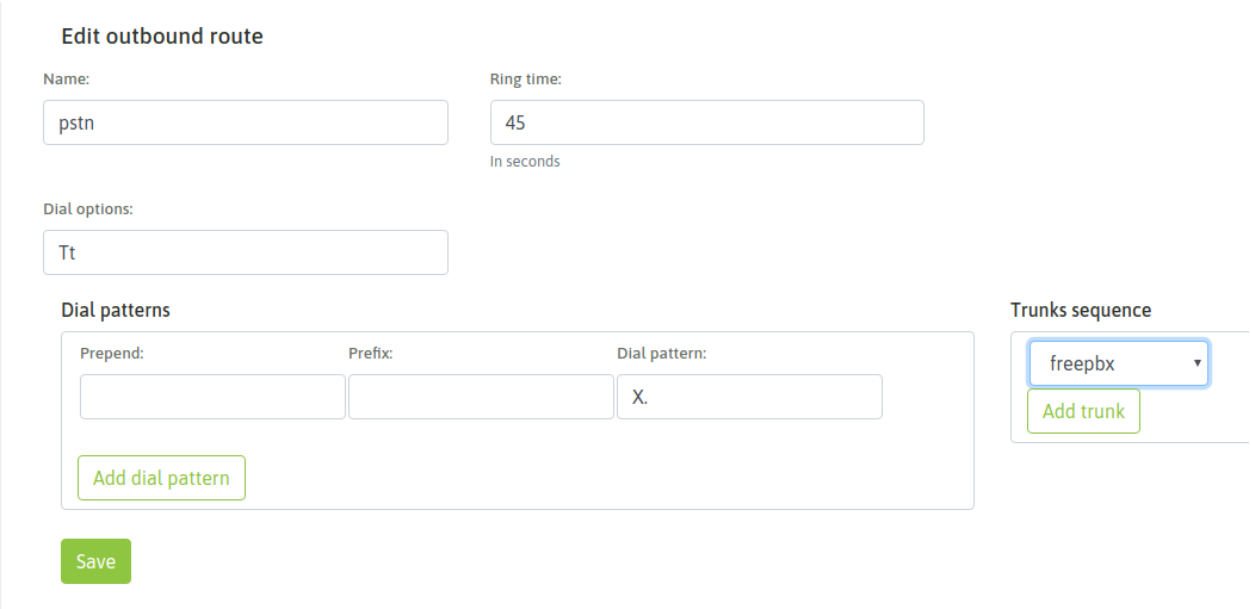


Los parámetros indicados en las figuras deben ser consistentes con los de su configuración.

11.2 Enrutamiento de llamadas entra ambos sistemas

Ruta saliente en OMniLeads

Podemos simplemente cambiar en la ruta ya generada el troncal por donde salir a la PSTN, utilizando el troncal generado en este laboratorio:

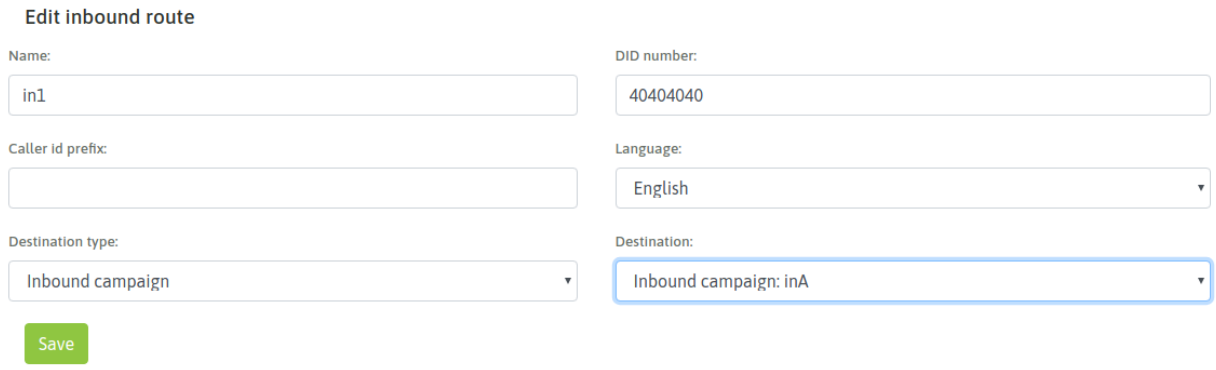


Por lo tanto cualquier número que se disque será encaminado hacia el PBX, dejando la decisión de enrutar hacia la PSTN o hacia algún recurso local las llamadas recibidas desde OMniLeads.

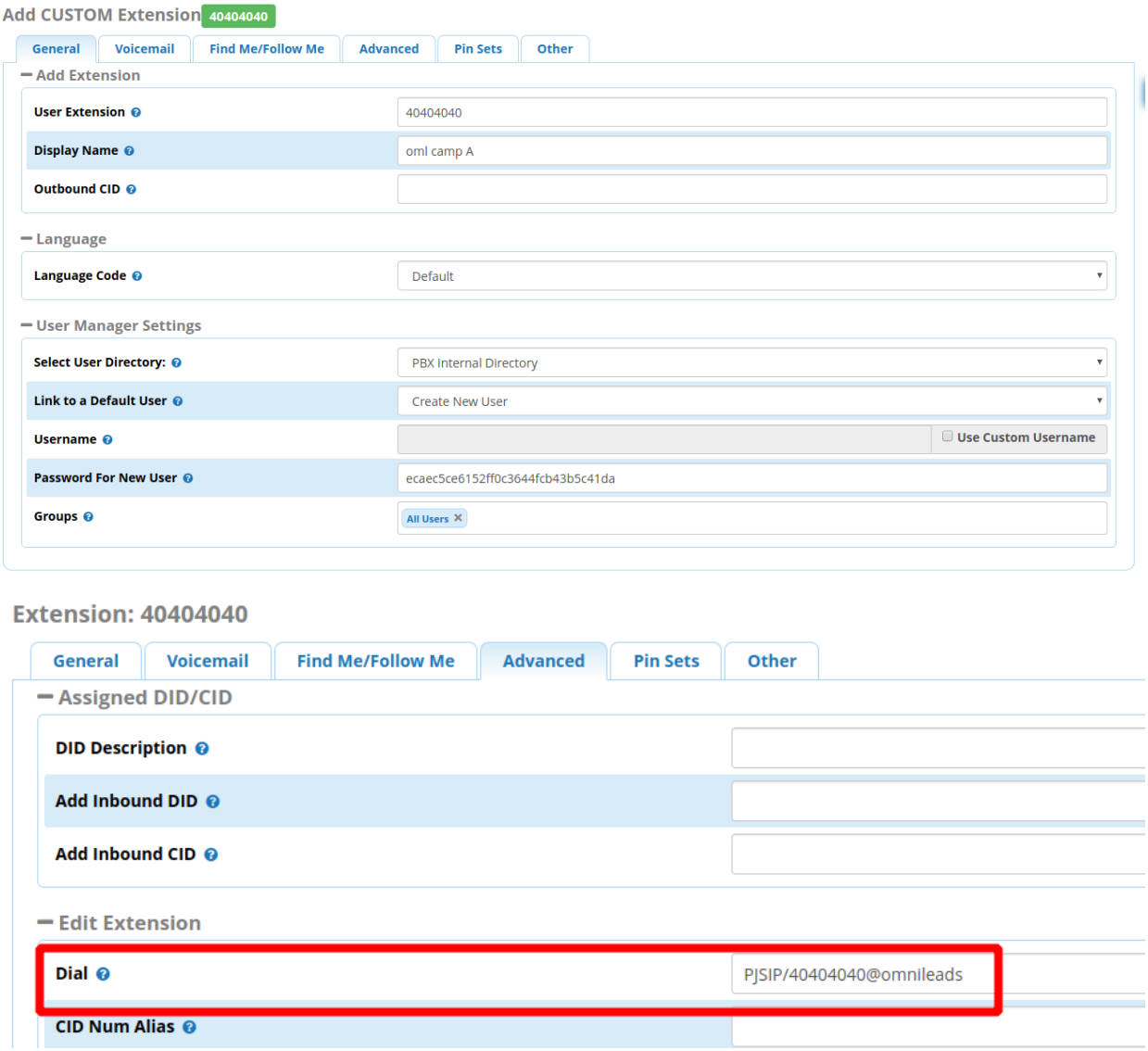
Encaminar llamadas desde el PBX hacia OMniLeads

Para poder marcar desde la PBX hacia destinos de OMniLeads es necesario relacionar cada *Ruta entrante* de OMniLeads con una *custom extension*.

Vamos a generar una extensión para la *Campaña Entrante A* que generamos unos laboratorios atrás. Para ello creamos una ruta entrante en OMniLeads que apunte a dicha campaña:



. . . y como contraparte la extensión en el PBX:



Nuevamente, los parámetros indicados en las figuras deben ser consistentes con los de su configuración.

11.3 Prueba de llamadas en ambos sentidos

- Registrar un softphone como una extensión SIP en el PBX, por otro lado ingresar a OMniLeads con un Agente asignado a la campaña entrante en cuestión.

Luego marcar desde el softphone el número de extensión que mapea ruta entrante de OMniLeads y la llamada deberá ingresar al agente de OMniLeads.

- Desde el agente, realizar una llamada «por fuera de campaña» hacia el número de Extensión asignado al Softphone y comprobar que la llamada haga ring en el softphone.

CAPÍTULO 12

Integración con CRM

En esta práctica vamos a explorar la integración entre OMniLeads y un sistema de gestión externo.

Las siguientes secciones de la Documentación Oficial cubren todos estos aspectos:

Integración con CRM

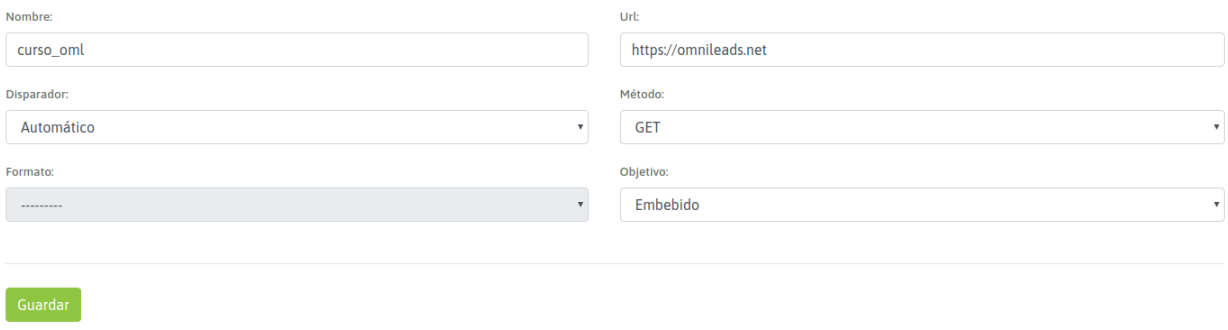
RESTful API

12.1 OML2CRM

Generar una interacción desde OMniLeads hacia una URL (servicio web o CRM) en cada llamada conectada entre un Agent y un Número externo (cliente / abonado).

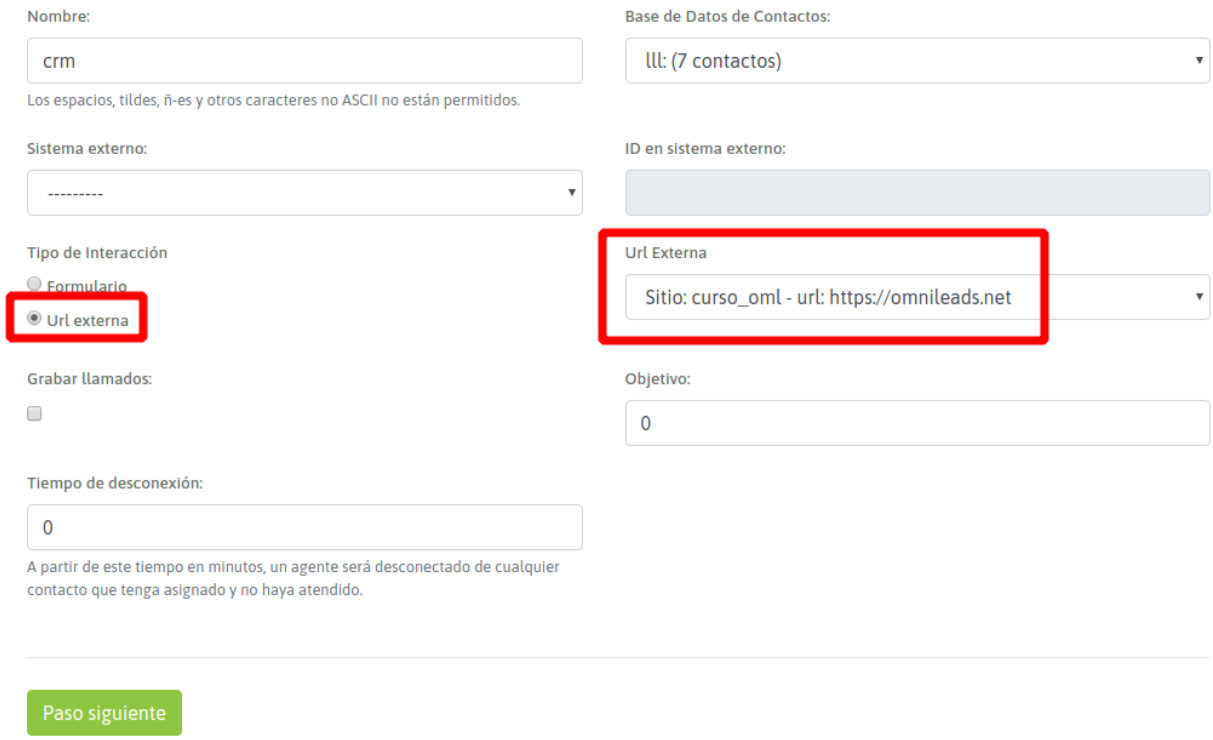
Activar un nuevo sistema externo

Crear un punto de interacción CRM utilizando la URL: crm-oml2crm-nuevo-sitio-externo.png y los parámetros de la figura

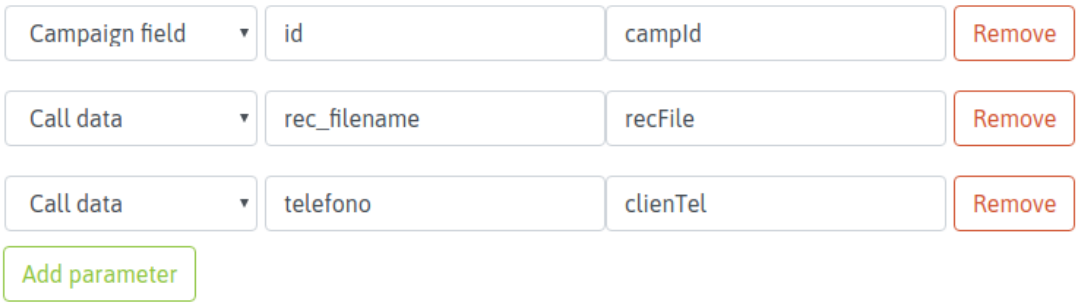


Configurar una campaña con interacción al sistema externo

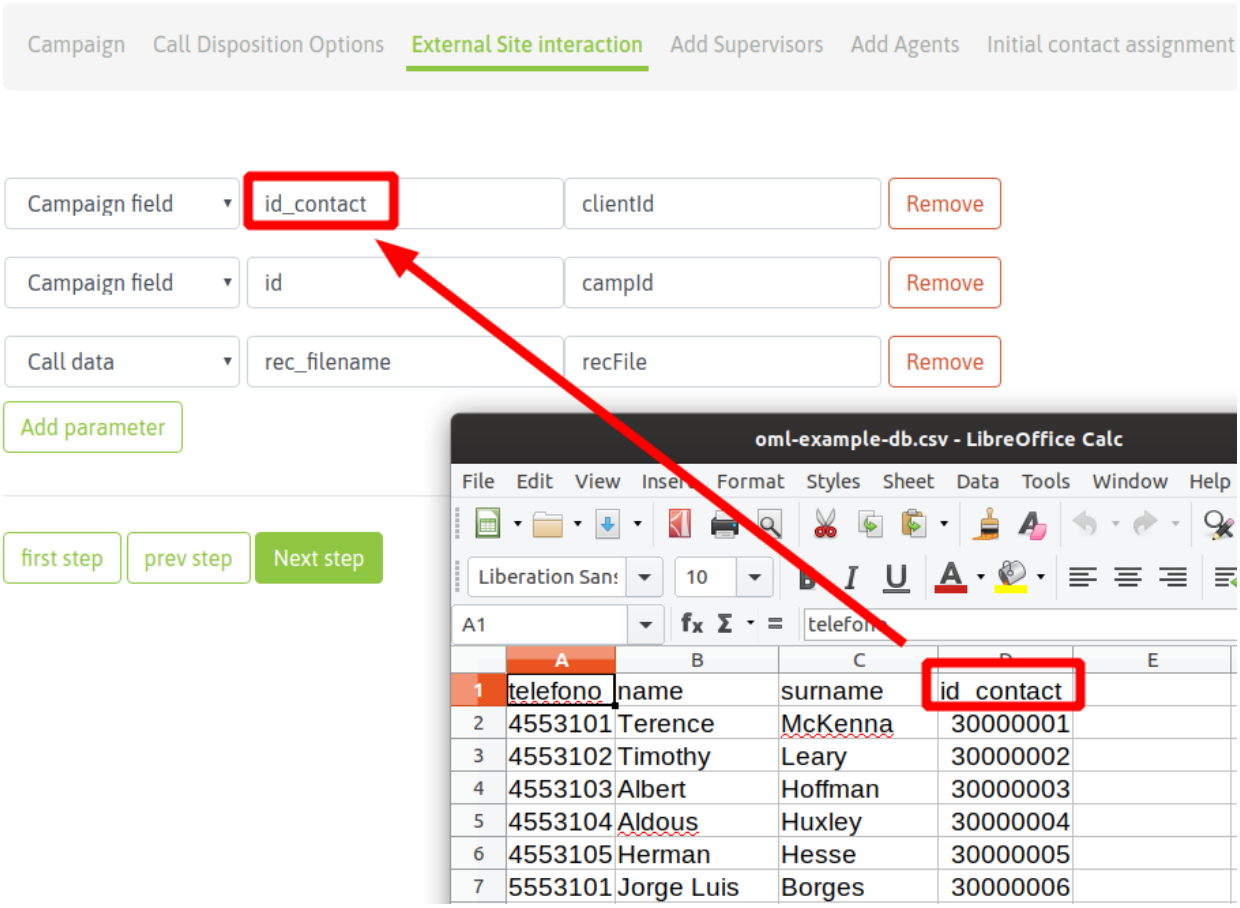
Ahora vamos a crear una campaña Preview utilizando el recurso de disparar una URL específica con parámetros de una llamada/campaña opcionalmente. Para ello generamos una campaña Preview y como primer punto seleccionamos el Sitio Externo con el cual vamos a trabajar en la campaña.



Finalmente poner atención en los parámetros que deseamos enviar, podemos implementar tal cual la figura:

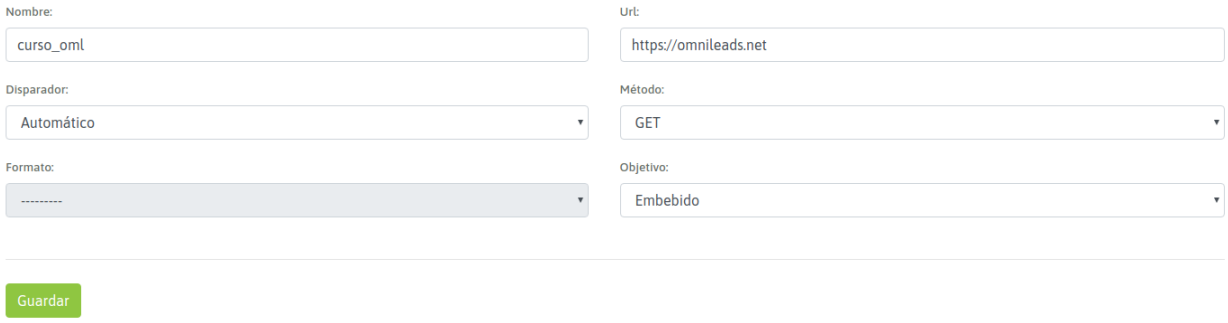


Tengamos en cuenta que se debe respetar tal cual el nombre de cada columna de la base de contactos, cuando se requiera invocar una columna de la misma a la hora de generar la llamada al URL.



Probar en las llamas

Ahora utilice cualquier agente para disparar llamadas preview desde nuestra nueva campaña.



CAPÍTULO 13

Gestiones del sysadmin

En los próximos pasos de este laboratorio vamos a estar trabajando con gestiones tíícas que debe llevar a cabo el sysadmin de OMniLeads.

La siguiente sección de la Documentación Oficial cubre todos estos aspectos.

13.1 Backup/Restore

Para este punto precisamos de una instancia de OMniLeads virgen sobre la cual vamos a llevar a cabo el restore. La instancia deberá tener la misma versión, misma IP/hostname y mismas contraseñas.

Sobre la instancia que estuvimos trabajando a lo largo de todo el curso, tomar un backup:

```
su omnileads -
cd /opt/omnileads/bin
./backup-restore.sh -b
```

El backup se genera en el directorio /opt/omnileads/backup.

Se deberá mover dicho archivo hacia la instancia donde se desea aplicar el restore (en el directorio /opt/omnileads/backup)

Ingresar vía SSH a la instancia de Restore y ejecutar los comandos:

```
su omnileads -
cd /opt/omnileads/bin
./backup-restore.sh -r nombre_del_archivo_de_backup
```

13.2 Actualizaciones

Para realizar una actualización, nos vamos a parar sobre el repositorio. Nos vamos a cambiar a la rama «develop» y allí ejecutar un post-install.

```
cd ominicontacto
git checkout develop
git pull origin develop
```

Una vez parados sobre la rama a implementar, debemos editar el archivo *inventory*.

Si estamos en modo «deployer & nodes»

```
##############################################################################
↪####
# If you are installing a prodenv (PE) AIO y bare-metal, change the IP and hostname␣
↪here #
##############################################################################
↪####
[prodenv-aio]
#localhost ansible_connection=local ansible_user=root #(this line is for self-hosted␣
↪installation)
192.168.95.100 ansible_ssh_port=22 ansible_user=root #(this line is for node-host␣
↪installation)
```

Es decir debemos editar la IP de nuestra instancia y luego ejecutar el script de deploy.

```
sudo ./deploy.sh -u
```

Si estamos en modo «self-hosted»

```
##############################################################################
↪####
# If you are installing a prodenv (PE) AIO y bare-metal, change the IP and hostname␣
↪here #
##############################################################################
↪####
[prodenv-aio]
localhost ansible_connection=local ansible_user=root #(this line is for self-hosted␣
↪installation)
#XXX.XXX.XXX.XXX ansible_ssh_port=22 ansible_user=root #(this line is for node-host␣
↪installation)
```

Es decir debemos descomentar la linea correspondiente y luego ejecutar el script de deploy.

```
./deploy.sh -u --iface=NIC_NAME


Cambios de contraseñas
**********************

Para llevar adelante modificaciones en las contraseñas:

* admin
* postgres
* asterisk AMI
* wombat dialer
* mysql
```

Debemos hacer un post-install habiendo antes modificado las contraseñas deseadas en el archivo de *inventory*.

Para este práctico, modificar la contraseña de *admin* y volver a ingresar al sistema utilizando dicha contraseña modificada.

13.3 Cambios de IP, hostname

Para ello se debe cambiar los parámetros de red deseados en el server que aloja la instancia de OMniLeads,reiniciar para que tome los valores. Luego se ejecuta un post-install.

Si estamos en modo «deployer & nodes»

```
##############################################################################
↪####
# If you are installing a prodenv (PE) AIO y bare-metal, change the IP and hostname␣
↪here #
##############################################################################
↪####
[prodenv-aio]
#localhost ansible_connection=local ansible_user=root #(this line is for self-hosted␣
↪installation)
192.168.95.100 ansible_ssh_port=22 ansible_user=root #(this line is for node-host␣
↪installation)
```

Es decir debemos descomentar e indicar la NUEVA IP de nuestra instancia y luego ejecutar el script de deploy.

```
sudo ./deploy.sh -u
```

Si estamos en modo «self-hosted»

```
##############################################################################
↪####
# If you are installing a prodenv (PE) AIO y bare-metal, change the IP and hostname␣
↪here #
##############################################################################
↪####
[prodenv-aio]
localhost ansible_connection=local ansible_user=root #(this line is for self-hosted␣
↪installation)
#XXX.XXX.XXX.XXX ansible_ssh_port=22 ansible_user=root #(this line is for node-host␣
↪installation)
```

Es decir debemos descomentar la linea correspondiente y luego ejecutar el script de deploy.

```
./deploy.sh -u --iface=NIC_NAME
```

En este punto, modificar la IP de la instancia de OMniLeads.