

cuan

MENSAJERÍA INSTANTÁNEA

EJERCICIO 4

Índice

Configuración de AWS.....	3
Crear un cluster.....	3
Asociar “Elastic IP” a la VPC	8
Modificar IAM Role de la Instancia	10
Crear Clúster.....	11
Registrar instancias externas.....	14
Reglas del firewall.	18
Definir Tareas:	19
Instalación de Prosody en EC2 – AWS.....	25
Limpieza Previa (Importante)	25
Puertos abiertos en la instancia en AWS.....	26
Instalación	27
Edición de nombres:	27
Configurar Prosody	28
Instalación de Certbot.	30
Generar los certificados.....	31
Después podemos añadir usuarios	32
Comprobación de puertos.	34
Cliente Pidgin	36
Añadir cuenta a Pidgin.	36
Ver certificados del servidor.	39
Mensaje instantáneo	40
Añadir amigo	42
Creación de Salas Abiertas	44
Sala Cerrada	48
Sala Cerrada – Solo Contraseña	52
Enviar archivo.....	55
Instalación de Funcionalidades:	57

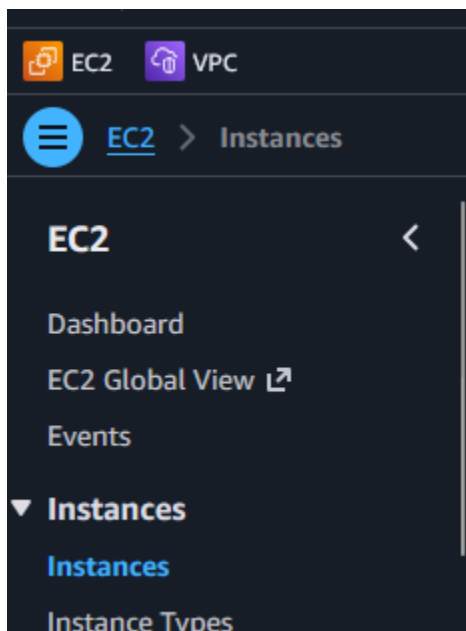
Plugin Installer	57
Nuevos Módulos y Componentes Instalados:	60
mod_admin_web	60
tombstones	63
mod_http_file_share	64
mod_mam	72
mod_checkcerts	73
mod_admin_shell	74
ANEXO:	76
Gajim en Windows:	76
Configuración final:	77

Configuración de AWS.

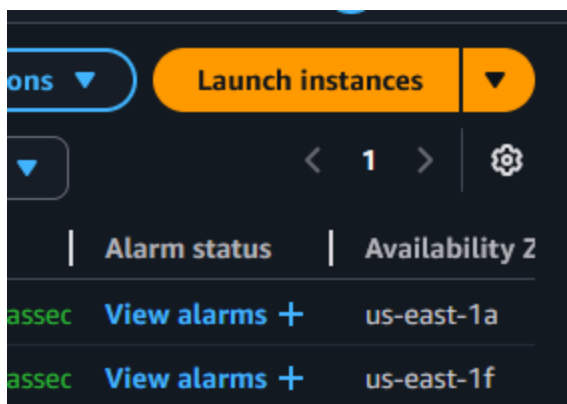
1. En pasos anteriores hemos creado una **VPC**.

Crear un cluster: Registrar una tarea y lanzar una instancia de contenedor en AWS ECS sobre EC2

Vamos a EC2 y en “Instances” → “Instances”.



Luego, en el panel de la derecha vamos le damos a “Launch Instances”.



Nombre que queramos:

Name and tags Info

Name

[Add additional tags](#)

Sistema: Elegimos Ubuntu y la versión LTS.

▼ Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

Deb

deb

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-0ecb62995f68bb549 (64-bit (x86)) / ami-01b9f1e7dc427266e (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Versión x64

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	
64-bit (x... ▼	ami-0ecb62995f68bb549	2025-10-22	ubuntu	<div>Verified provider</div>

Tipo de instancia: elige los recursos necesarios según la tarea. En nuestro caso elegimos la t3.small

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.small Free tier eligible
Family: t3 2 vCPU 2 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0518 USD per Hour
On-Demand Linux base pricing: 0.0208 USD per Hour
On-Demand RHEL base pricing: 0.0496 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0243 USD per Hour
On-Demand Windows base pricing: 0.0392 USD per Hour

☒ All generations
[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software


Key par login: Para la conexión a la máquina. Elige “vockey” porque ya tiene la configuración para SSH.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name – required

vockey ▼

 [Create new key pair](#)

Opciones de red:

- Network: Elegimos la VPC que creamos en pasos anteriores.
- Common security groups: Elegimos el que se crea por defecto con el VPC.

▼ Network settings [Info](#) [Edit](#)

Network | [Info](#)
vpc-0073baba5401276f7

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable

Firewall (security groups) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups | [Info](#)
Select security groups ▼

default sg-00f0a12f377afd310 ✕
VPC: vpc-0073baba5401276f7

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

O también podemos usar un grupo de seguridad propio. **Creo que este caso elegimos esta opción.**

Firewall (security groups) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0 ▼

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

Almacenamiento: 8 GB.

▼ **Configure storage** Info

Advanced

1x 8 GiB gp3 ▼ Root volume, 3000 IOPS, Not encrypted

Add new volume

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

🕒 Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

🔄

0 x File systems

Edit

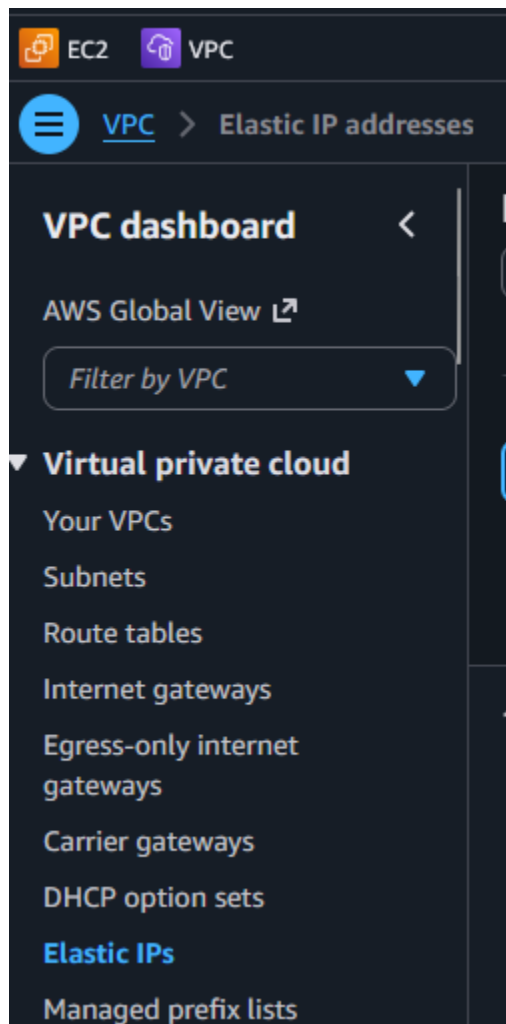
Una vez terminado, le damos a “Launch Instance”.

Launch instance

📄 Preview code

Asociar “Elastic IP” a la VPC:

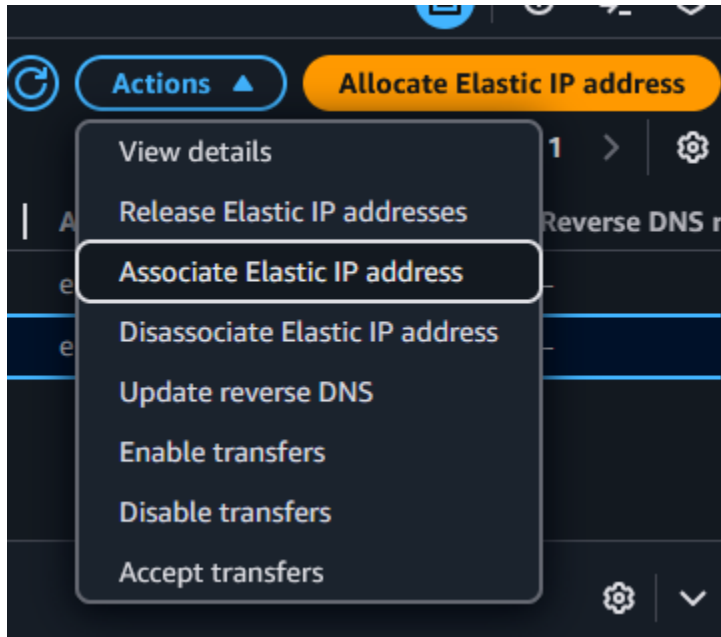
Nos vamos a VPC → “Elastic IPs”



Seleccionamos la IP Elástica que vamos a asociar.



Y en el panel de la derecha le damos a “Associate Elastic IP address”.



Seleccionas la instancia a la que la quieres otorgar y también le indicas una IP Privada.

Associate Elastic IP address [Info](#)

Choose the instance or network interface to associate to this Elastic IP address (100.48.114.98)

Elastic IP address: 100.48.114.98

Resource type
Choose the type of resource with which to associate the Elastic IP address.

☒ Instance
☐ Network interface

Warning: If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the instance will retain its primary private IP address. If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance
i-095252e73ecee94c1

Private IP address
The private IP address with which to associate the Elastic IP address.
172.31.75.85

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
☒ Allow this Elastic IP address to be reassociated

Modificar IAM Role de la Instancia:

Seleccionas la instancia y nos vamos a

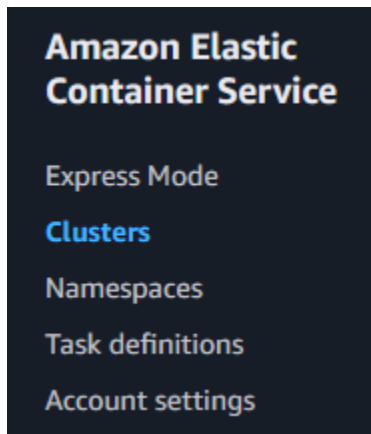


Selecciona

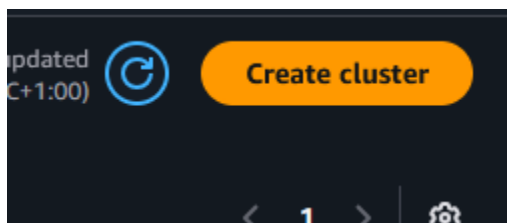


Crear Clúster.

En ECS (Elastic Container Service) nos vamos a “Clúster”.



Crear Clúster



Nombre:

Create cluster [Info](#)

An Amazon ECS cluster groups together tasks, and services, and allows for shared capacity and common configurations. must belong to a cluster.

Cluster configuration

Cluster name

Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

► **Service Connect defaults - *optional***

En Infraestructura

▼ Infrastructure - *advanced* [Info](#)

Configure the manner of obtaining compute resources that will be used to host your application.

Select a method of obtaining compute capacity

Your cluster is automatically configured for AWS Fargate (serverless), but you may choose to add Amazon EC2 instances (servers).

☐ **Fargate only**

Serverless - you don't think about creating or managing servers. Great for most common workloads.

☐ **Fargate and Managed Instances**

Managed instances - Amazon ECS will manage patching and scaling on your behalf while giving you configurability about the types of instances. Great for more advanced workloads.

☒ **Fargate and Self-managed instances**

Self-managed instances - you must ensure the instances are patched and scaled properly, and you have full control over the instances.

Auto Scaling group (ASG) [Info](#)

☒ Create a new Auto Scaling group - *advanced*

☐ Use an existing Auto Scaling group

Container instance Amazon Machine Image (AMI)

Choose the Amazon ECS-optimized AMI for your instance.

Amazon Linux 2023

EC2 instance type

Choose based on the workloads you plan to run on this cluster.

t3a.small
x86_64
2 vCPU 2 GiB Memory

EC2 instance role

An instance role is used by Amazon EC2 instances to make AWS API requests. If you don't already have an instance IAM role created, we can create one for you.

LabInstanceProfile
arn:aws:iam::905418476068:instance-profile/LabInstanceProfile

Desired capacity

Specify the number of instances to launch in your cluster.

Minimum **Maximum**

0 5

SSH Key pair

If you do not specify a key pair, you can't connect to the instances via SSH unless you choose an AMI that is configured to allow users another way to log in.

vockey [Create a new key pair](#)

Root EBS volume size

You can increase the size of the root EBS volume to allow for greater image and container storage.

30

A min of 30 GiB and a max of 16,384 GiB is allowed.

No tocamos en este caso los “Network Settings”.

► Network settings

The default VPC, subnets, and security groups are selected by default.

Y le damos a crear.

Clusters (1) [Info](#)

Last updated January 3, 2026, 18:53 (UTC+1:00)

[Create cluster](#)

By default, we only load up to 1000 clusters at a time.

< 1 > [Settings](#)

Cluster	Services	Tasks	Container instances	CloudWatch monitoring
ECS-csaV3	0	No tasks running	0 EC2 1 External	<input checked="" type="radio"/> Default

Registrar instancias externas.

ECS → Clústers → Selecciona el Clúster Creado → Infraestructura → Registrar Instancias Externas.

The screenshot shows the Amazon Elastic Container Service (ECS) console interface. The left sidebar contains navigation links for Express Mode, Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main content area is titled 'Infrastructure' and includes tabs for Services, Tasks, Metrics, Scheduled tasks, Configuration, Event history, and Tags. The 'Capacity providers' section shows a table with three providers: Fargate, Fargate Spot, and Infra-ECS-Cluster-ECS-csaV3-1fe7dc... (selected). The 'Container instances' section is empty, and a red arrow points to the 'Register external instances' button.

Capacity provider	Scaling type	Provisioning model	Update status	Update status reason
Fargate	Fargate	-	-	-
Fargate Spot	Fargate Spot	-	-	-
Infra-ECS-Cluster-ECS-csaV3-1fe7dc...	EC2 Auto Scaling	-	-	-

Container instance	Instance type	Status	Statu...	Type	Instance ID	Capacit...	Availability zo...	Running tasks...	CPU availa
--------------------	---------------	--------	----------	------	-------------	------------	--------------------	------------------	------------

No container instances

Rol de la Instancia “LabRole”. Y le damos a generar comando de registro.

Register external instances

Registering external instances with your cluster has a pricing impact. [Learn more](#)

Activation key duration (in days)

The activation key pair is used to securely register your external instances with Amazon ECS.

1

Activation key duration (in days) should be less than or equal to 30.

Number of instances

The maximum number of external instances you want to register with this activation.

1

Valid range: Minimum value of 1. Maximum value of 1000.

Instance role

This role is required by external instances to communicate with AWS APIs on your behalf. If you don't have an external instance IAM role, we can create one for you. [Learn more](#)

LabRole

Generate registration command

Close

Nos saldrán dos comandos: uno para Linux y otro para Windows. Cópialos porque nos hará falta más adelante.

Linux Command

Complete the registration process by running this command as root on your external instances. returned. For external instances with GPUs, add the `--enable-gpu` flag to the command to support GPU workloads.

```
curl --proto "https" -o "/tmp/ecs-anywhere-install.sh" "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh" && bash /tmp/ecs-anywhere-install.sh --region "us-east-1" --cluster "ECS-csaV3" --activation-id "1ae02d82-022d-4d14-bcd9-9a94dbd8bf7f" --activation-code "ZhmlOgKseBRBe9PIOmix"
```

✓ Command copied to clipboard

Copy

Windows Command

Complete the registration process by running this command as Administrator on your external instances. If the command is not run as Administrator, an error is returned.

```
Invoke-RestMethod -URI "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install.ps1" -OutFile "ecs-anywhere-install.ps1"; .\ecs-anywhere-install.ps1 -Region "us-east-1" -Cluster "ECS-csaV3" -ActivationID "1ae02d82-022d-4d14-bcd9-9a94dbd8bf7f" -ActivationCode "ZhmlOgKseBRBe9PIOmix"
```

Copy

Close

Ahora debemos conectarnos a la instancia de EC2 que creamos antes. En nuestro caso Diego le ha otorgado ya un nombre en el DNS. Debemos usar el archivo “labsuser.pem” que nos descargamos antes. Nos posicionamos en el directorio donde esté el archivo usando el terminal y ejecutamos:

```
ssh -i .\labuser.pem ubuntu@im07.institutodh.net
```

(En mi caso es “labuser.pem”).

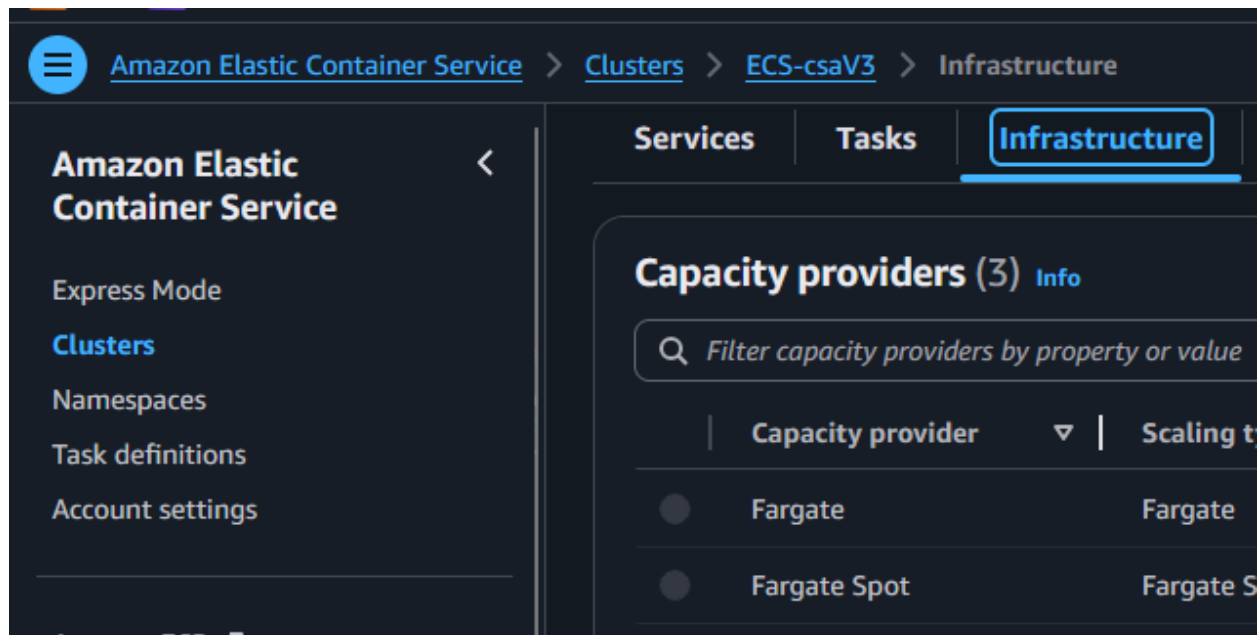
Una vez dentro, ejecutamos como ROOT el comando que copiamos previamente, en nuestro caso la versión para Linux.

```
ubuntu@ip-172-31-75-85:~$ sudo su
root@ip-172-31-75-85:/home/ubuntu# curl --proto "https" -o "/tmp/ecs-anywhere-install.sh" "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh" && bash /tmp/ecs-anywhere-install.sh --region "us-east-1" --cluster "ECS-csaV3" --activation-id "1ae02d82-022d-4d14-bcd9-9a94dbd8bf7f" --activation-code "ZhmlOgKseBRBe9PIOmix"
```

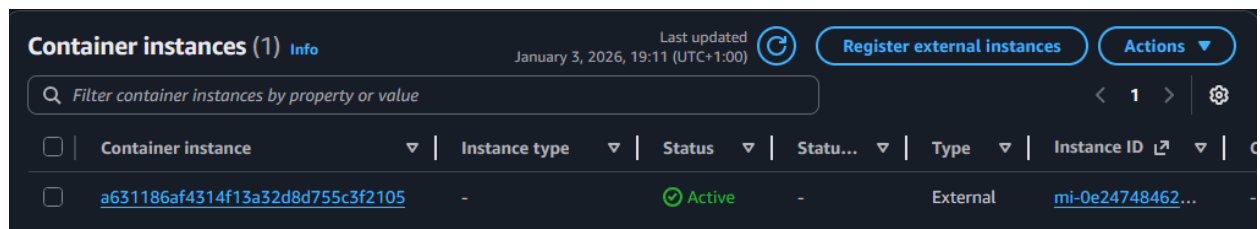
```
#####
This script installed three open source packages that all use Apache License 2.0.
You can view their license information here:
- ECS Agent https://github.com/aws/amazon-ecs-agent/blob/master/LICENSE
- SSM Agent https://github.com/aws/amazon-ssm-agent/blob/master/LICENSE
- Docker engine https://github.com/moby/moby/blob/master/LICENSE
#####

root@ip-172-31-75-85:/home/ubuntu# |
```

Ahora ve a



Y en la parte de abajo verás que está la instancia creada:



Reglas del firewall.

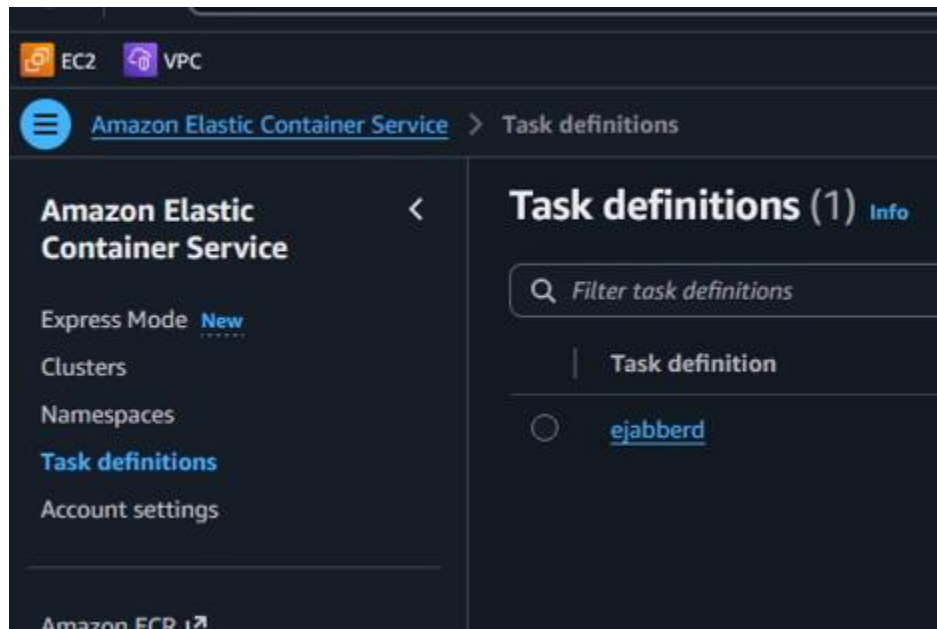
Se aplican a la instancia → Security groups → Grupo de seguridad → Inbound/Outbound Rules.

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows the navigation menu with categories like EC2, Images, and Elastic Block Store. The main content area is titled 'Operator' and shows the instance ID 'i-095252e73ecee94c1'. The 'Security' tab is selected, showing 'Security details' and 'Inbound rules'. The 'Inbound rules' section contains a table with one rule.

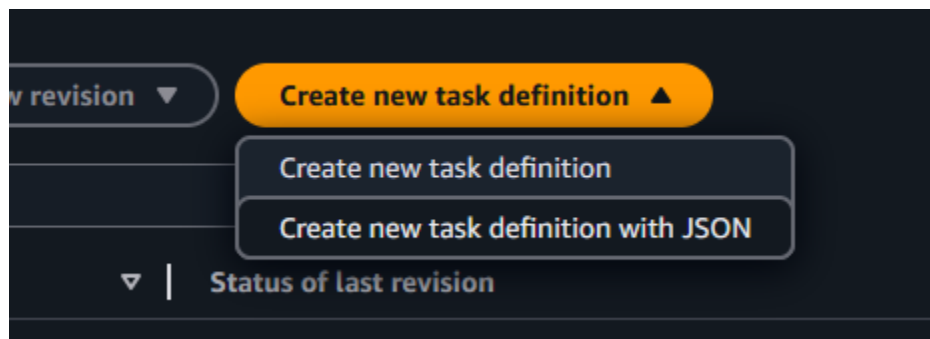
Name	Security group rule ID	Port range
-	sgr-00dc2c30fedfdc025	22

Definir Tareas:

ECS → Task Definitions



Y le damos a



Ejemplo de configuración de JSON para Prosody:

```
{
  "family": "prosody-v1",
  "containerDefinitions": [
    {
      "cpu": 0,
      "dnsSearchDomains": [
        "institutodh.net"
      ],
      "dnsServers": [
        "1.1.1.1"
      ],
      "environment": [
        {
          "name": "PROSODY_DOMAIN",
          "value": "im07.institutodh.net"
        }
      ],
      "essential": true,
      "hostname": "im07",
      "image": "prosodyim/prosody:13.0",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/prosody-v1",
          "awslogs-create-group": "true",
          "awslogs-region": "us-east-1",
          "awslogs-stream-prefix": "ecs"
        }
      },
      "mountPoints": [
        {
          "containerPath": "/etc/prosody",
          "readOnly": false,
          "sourceVolume": "prosody-conf"
        },
        {
          "containerPath": "/var/lib/prosody",
          "readOnly": false,
          "sourceVolume": "prosody-data"
        }
      ],
      "name": "prosody-ct",
```

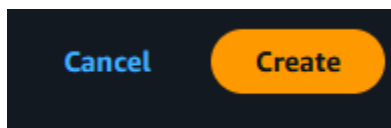
```
"portMappings": [  
  {  
    "containerPort": 5000,  
    "hostPort": 5000,  
    "name": "file-transfer-proxy",  
    "protocol": "tcp"  
  },  
  {  
    "containerPort": 5222,  
    "hostPort": 5222,  
    "name": "client-connections",  
    "protocol": "tcp"  
  },  
  {  
    "containerPort": 5269,  
    "hostPort": 5269,  
    "name": "s2s-connections",  
    "protocol": "tcp"  
  },  
  {  
    "containerPort": 5280,  
    "hostPort": 5280,  
    "name": "http-private",  
    "protocol": "tcp"  
  },  
  {  
    "containerPort": 5281,  
    "hostPort": 5281,  
    "name": "https-public",  
    "protocol": "tcp"  
  },  
  {  
    "containerPort": 5347,  
    "hostPort": 5347,  
    "name": "external-components",  
    "protocol": "tcp"  
  },  
  {  
    "containerPort": 5582,  
    "hostPort": 5582,  
    "name": "telnet-console",  
    "protocol": "tcp"  
  }  
],
```

```

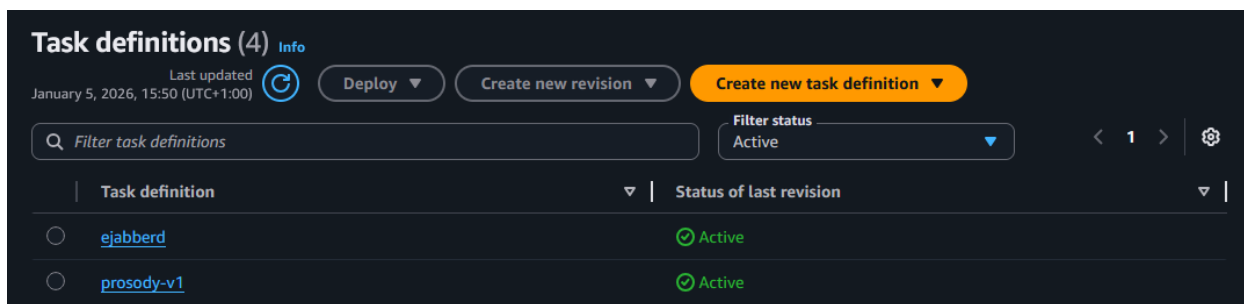
    "systemControls": [],
    "volumesFrom": []
  },
  ],
  "taskRoleArn": "arn:aws:iam::905418476068:role/LabRole",
  "executionRoleArn": "arn:aws:iam::905418476068:role/LabRole",
  "networkMode": "bridge",
  "volumes": [
    {
      "host": {
        "sourcePath": "/home/ubuntu/prosody/conf"
      },
      "name": "prosody-conf"
    },
    {
      "host": {
        "sourcePath": "/home/ubuntu/prosody/data"
      },
      "name": "prosody-data"
    }
  ],
  "requiresCompatibilities": [
    "EC2"
  ],
  "cpu": "1600",
  "memory": "1600",
  "runtimePlatform": {
    "cpuArchitecture": "X86_64",
    "operatingSystemFamily": "LINUX"
  },
  "enableFaultInjection": false
}

```

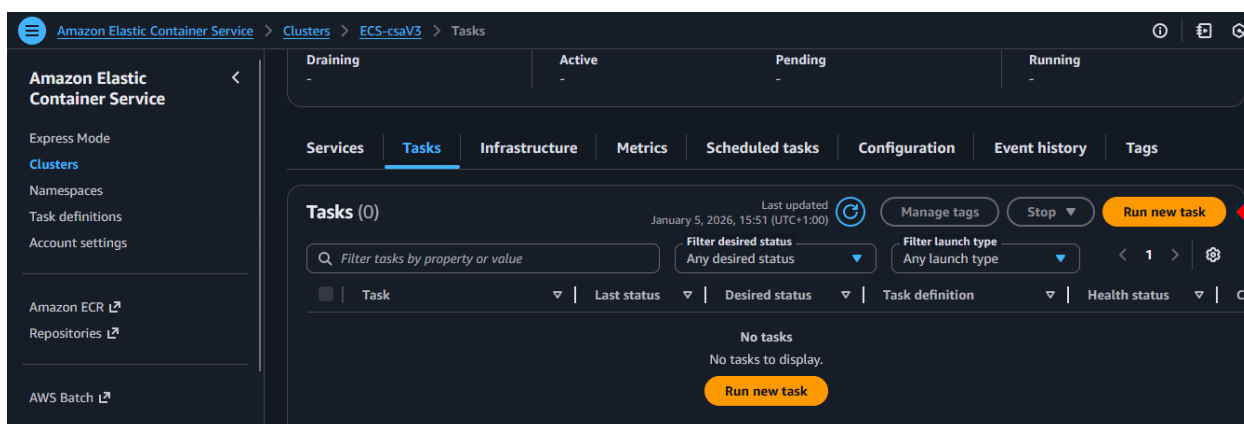
Le damos a “Crear”.



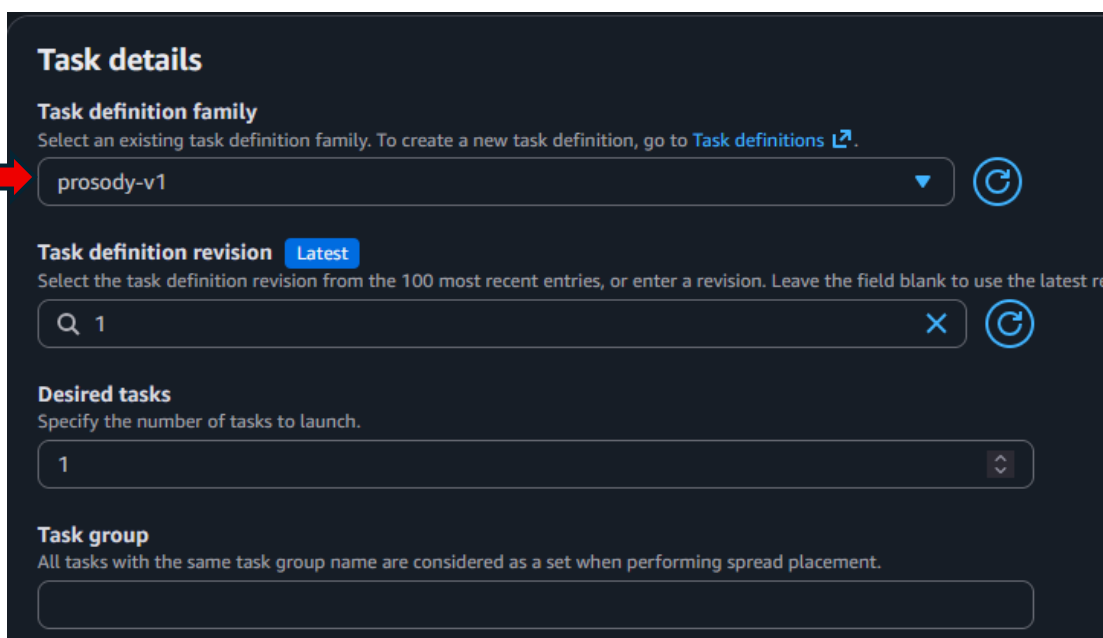
Después podemos ver que está creada.



Ahora tenemos que irnos a Clúster → Seleccionamos el clúster → Tareas → Ejecutar Nueva Tarea.



Seleccionamos la tarea que hemos definido antes.



Y hasta ahí hemos llegado con el ECS, porque me dan una serie de problemas que no he sabido solucionar. Por eso lo he hecho en EC2.

Instalación de Prosody en EC2 – AWS.

Nos conectamos a la instancia:

```
ssh -i .\labuser.pem ubuntu@im07.institutodh.net
```

Limpieza Previa (Importante)

Antes de empezar con el paso 1, asegúrate de que no haya restos de intentos anteriores o del agente de ECS que puedan chocar con los puertos:

```
sudo systemctl stop ecs
```

```
sudo docker stop prosody || true
```

```
sudo docker rm prosody || true
```

Puertos abiertos en la instancia en AWS.

Como hemos visto antes, los puertos de las instancias se configuran en:

EC2 → Selecciona la instancia → Security → Security Groups → Inbound/Outbound Rules.

En nuestro caso solo hemos tenido que tocar las Inbound.

Name	Security group rule ID	Port range	Protocol	Source
—	sgr-0d2cded84d566d593	All	ICMP	0.0.0.0/0
—	sgr-017f181ec96b8df25	80	TCP	0.0.0.0/0
—	sgr-05903d4f4d35ccff3	5281	TCP	0.0.0.0/0
—	sgr-0454d05d9fdbad170	22	TCP	0.0.0.0/0
—	sgr-057bed7c704a26ebf	5280	TCP	0.0.0.0/0
—	sgr-0f1e093fa68c9488f	5582	TCP	0.0.0.0/0
—	sgr-00dc2c30fedfdc025	5222	TCP	0.0.0.0/0
—	sgr-0b838755ff6aedef46	5000	TCP	0.0.0.0/0
—	sgr-081e222480628b4b5	5347	TCP	0.0.0.0/0
—	sgr-02ab7b979446fa80a	5269	TCP	0.0.0.0/0

¿Qué puertos debemos abrir?

En la página oficial¹ se indican los puertos que deben usarse.

Default ports

Here is an overview of default ports and the respective services:

port	interfaces	service
5000/tcp	public	File transfer proxy
5222/tcp	public	Client connections
5269/tcp	public	Server-to-server connections
5280/tcp	private ¹	HTTP
5281/tcp	public	HTTPS
5347/tcp	private	External components
5582/tcp	private	Telnet console

¹ <https://prosody.im/doc/ports>

Instalación

```
sudo apt update
```

```
sudo apt install prosody
```

Edición de nombres:

```
"sudo nano /etc/hosts"
```

```
127.0.0.1    localhost
```

```
127.0.1.1    im07.institutodh.net im07
```

```
:::1         localhost ip6-localhost ip6-loopback
```

```
GNU nano 7.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    im07.institutodh.net im07
:::1         localhost ip6-localhost ip6-loopback

# The following lines are desirable for IPv6 capable hosts
:::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

```
sudo nano /etc/hostname
```

```
o
```

```
sudo hostnamectl set-hostname im07
```

```
GNU nano 7.2
im07
```

Configurar Prosody

Para ello debemos usar el archivo “**prosody.cfg.lua**” que se ha creado en “**/etc/prosody/**”.

Configuración: <https://prosody.im/doc/configure>

Certificados: <https://prosody.im/doc/certificates>

ATENCIÓN: Esta es una configuración inicial. Más adelante se harán modificaciones.

```
admins = { "admin@im07.institutodh.net" }
plugin_paths = { "/usr/lib/prosody/modules" }
modules_enabled = {
    "roster"; -- Allow users to have a roster. Recommended ;)
    "saslauth"; -- Authentication for clients and servers. Recommended if you want
to log in.
    "tls"; -- Add support for secure TLS on c2s/s2s connections
    "dialback"; -- s2s dialback support
    "disco"; -- Service discovery

    "carbons"; -- Keep multiple clients in sync
    "pep"; -- Enables users to publish their avatar, mood, activity, playing music
and more
    "private"; -- Private XML storage (for room bookmarks, etc.)
    "blocklist"; -- Allow users to block communications with other users
    "vcard4"; -- User profiles (stored in PEP)
    "vcard_legacy"; -- Conversion between legacy vCard and PEP Avatar, vcard
    "limits"; -- Enable bandwidth limiting for XMPP connections

    "version"; -- Replies to server version requests
    "uptime"; -- Report how long server has been running
    "time"; -- Let others know the time here on this server
    "ping"; -- Replies to XMPP pings with pongs
    "register"; -- Allow users to register on this server using a client and change
passwords
    -- Admin interfaces
    "admin_adhoc"; -- Allows administration via an XMPP client that supports ad-
hoc commands

    -- Other specific functionality
    "posix"; -- POSIX functionality, sends server to background, enables syslog,
etc.
}

modules_disabled = {
```

```
}

```

allow_registration = false – Evita que los usuarios creen cuentas desde Pidgin, en teoría. El módulo "register" debe estar habilitado.

daemonize = true; -- Si lo dejas en "false", cuando cierres la terminal SSH, el servidor de Prosody se detendrá.

pidfile = "/run/prosody/prosody.pid";

c2s_require_encryption = true -- SEGURIDAD: Obliga a usar cifrado entre cliente y servidor

s2s_require_encryption = true -- SEGURIDAD: Obliga a usar cifrado entre servidores

s2s_secure_auth = true

allow_unencrypted_plain_auth = false

-- Límite velocidad conexión

limits = {

c2s = {

rate = "50kb/s";

};

s2sin = {

rate = "100kb/s";

};

}

authentication = "internal_hashed"

archive_expires_after = "2w" -- Remove archived messages after 1 week

log = {

-- Log files (change 'info' to 'debug' for debug logs):

debug = "/var/log/prosody/prosody.log";

error = "/var/log/prosody/prosody.err";

-- Syslog:

{ levels = { "error" }; to = "syslog"; };

}

certificates = "certs"

VirtualHost "im07.institutodh.net"

ssl = {

key = "/etc/prosody/certs/im07.institutodh.net.key";

certificate = "/etc/prosody/certs/im07.institutodh.net.crt";

}

Component "conference.im07.institutodh.net" "muc"

modules_enabled = { "muc_mam" }

Include "conf.d/*.cfg.lua"

Instalación de Certbot.

1. Preparación del entorno Snap:

Instala los archivos fundamentales para que el sistema Snap funcione correctamente.

sudo snap install core

Asegura que el "motor" de Snap esté actualizado a la última versión para evitar errores de compatibilidad durante la instalación de aplicaciones.

sudo snap refresh core

2. Limpieza de versiones antiguas

sudo apt-get remove certbot

3. Instalación de Certbot vía Snap

sudo snap install --classic certbot

4. Creación del enlace simbólico

sudo ln -s /snap/bin/certbot /usr/bin/certbot

Este comando crea un "acceso directo" (enlace simbólico) en /usr/bin/. Esto permite que, cuando escribas simplemente certbot en la terminal, el sistema sepa exactamente dónde encontrar el programa y lo ejecute sin errores.

Generar los certificados.

Previamente se debe haber puesto en Godaddy el dominio `imXX.institutodh.net` y la IP elástica que hemos adjuntado a la instancia en AWS.

1. Generación del certificado:

sudo certbot certonly --standalone

`certonly`: Le indica a Certbot que solo obtenga el certificado.

`--standalone`: Este es el "modo autónomo".

2. Verificación de los archivos creados

sudo ls -ltra /etc/letsencrypt/live/im07.institutodh.net

ATENCIÓN: Asegúrate de que no haya otro servicio usando el **puerto 80** y también de que has abierto dicho puerto en “**Security Groups**” (AWS) de la instancia (en la regla de Inbounds). Esto se debe a que Certbot tiene que levantar un servidor temporal para validar tu dominio.

▼ Inbound rules				
<input type="text" value="Filter rules"/> < 1				
Name	Security group rule ID	Port range	Protocol	Source
–	sgr-0d2cded84d566d593	All	ICMP	0.0.0.0/0
–	sgr-017f181ec96b8df25	80	TCP	0.0.0.0/0
–	sgr-05903d4f4d35ccff3	5281	TCP	0.0.0.0/0
–	sgr-0454d05d9fdbad170	22	TCP	0.0.0.0/0
–	sgr-057bed7c704a26ebf	5280	TCP	0.0.0.0/0
–	sgr-00dc2c30fedfdc025	5222	TCP	0.0.0.0/0
–	sgr-0b838755ff6aedf46	5000	TCP	0.0.0.0/0
–	sgr-02ab7b979446fa80a	5269	TCP	0.0.0.0/0

En la web oficial se indica que Prosody por norma general no puede usar los certificados cuando están en el directorio de letsencrypt² por temas de seguridad. Por eso debemos usar este comando para importarlos:

/usr/bin/prosodyctl --root cert import /etc/letsencrypt/live

² <https://prosody.im/doc/letsencrypt>

Podemos crear un script con dicho comando para que se ejecute con cada renovación de las credenciales.

También podemos asegurarnos de que prosody pueda usarlos con:

```
sudo chown prosody:prosody /etc/prosody/certs/im07.institutodh.net.*
```

```
sudo chmod 640 /etc/prosody/certs/im07.institutodh.net.*
```

Después reiniciamos el servicio y comprobamos el estatus:

```
sudo systemctl restart prosody
```

```
sudo systemctl status prosody
```

Después podemos añadir usuarios³:

```
sudo prosodyctl adduser admin@im07.institutodh.net
```

Borrar usuarios⁴:

```
sudo prosodyctl deluser admin@im07.institutodh.net
```

Cambiarles la contraseña:

```
sudo prosodyctl passwd admin@im07.institutodh.net
```

³ https://prosody.im/doc/creating_accounts

⁴ <https://prosody.im/doc/prosodyctl>

¿Cómo podemos saber que usuarios tenemos?

- Método 1⁵:

ls -l /var/lib/prosody/*/accounts/*

```
root@im07:/home/ubuntu# ls -l /var/lib/prosody/*/accounts/*
-rw-r----- 1 prosody prosody 218 Jan  5 11:22 /var/lib/prosody/im07%2einstitutodh%2enet/accounts/admin.dat
-rw-r----- 1 prosody prosody 218 Jan  5 11:23 /var/lib/prosody/im07%2einstitutodh%2enet/accounts/cristobal.dat
-rw-r----- 1 prosody prosody 218 Jan  6 10:44 /var/lib/prosody/im07%2einstitutodh%2enet/accounts/gaben.dat
root@im07:/home/ubuntu# |
```

- Método 2:

Podemos usar “prosodyctl shell user list (HOST)”. Ejemplo:

prosodyctl shell user list im07.institutodh.net

```
root@im07:/home/ubuntu# prosodyctl shell user list im07.institutodh.net
cristobal@im07.institutodh.net
admin@im07.institutodh.net
gaben@im07.institutodh.net
putin@im07.institutodh.net
donald@im07.institutodh.net
movil@im07.institutodh.net
OK: Showing all 6 users
root@im07:/home/ubuntu# |
```

⁵ <https://wiki.archlinux.org/title/Prosody>

Comprobación de puertos.

ss -ltnp

```
root@im07:/etc/prosody# ss -ltnp
State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
LISTEN 0 128 0.0.0.0:5269 0.0.0.0:* users:(("lua5.4",pid=28046,fd=10))
LISTEN 0 4096 127.0.0.54:53 0.0.0.0:* users:(("systemd-resolve",pid=370,fd=17))
LISTEN 0 4096 0.0.0.0:22 0.0.0.0:* users:(("sshd",pid=1496,fd=3),("systemd",pid=1,fd=235))
LISTEN 0 128 0.0.0.0:5222 0.0.0.0:* users:(("lua5.4",pid=28046,fd=7))
LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:* users:(("systemd-resolve",pid=370,fd=15))
LISTEN 0 128 [::]:5269 [::]:* users:(("lua5.4",pid=28046,fd=11))
LISTEN 0 4096 [::]:22 [::]:* users:(("sshd",pid=1496,fd=4),("systemd",pid=1,fd=236))
LISTEN 0 128 [::]:5222 [::]:* users:(("lua5.4",pid=28046,fd=8))
root@im07:/etc/prosody#
```

sudo lsof -i -P -n

```
root@im07:/etc/prosody# sudo lsof -i -P -n
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
systemd 1 root 235u IPv4 6397 0t0 TCP *:22 (LISTEN)
systemd 1 root 236u IPv6 6399 0t0 TCP *:22 (LISTEN)
systemd-r 370 systemd-resolve 14u IPv4 3748 0t0 UDP 127.0.0.53:53
systemd-r 370 systemd-resolve 15u IPv4 3749 0t0 TCP 127.0.0.53:53 (LISTEN)
systemd-r 370 systemd-resolve 16u IPv4 3750 0t0 UDP 127.0.0.54:53
systemd-r 370 systemd-resolve 17u IPv4 3751 0t0 TCP 127.0.0.54:53 (LISTEN)
systemd-n 523 systemd-network 23u IPv4 128799 0t0 UDP 172.31.75.85:68
chronyd 665 _chrony 5u IPv4 6921 0t0 UDP 127.0.0.1:323
chronyd 665 _chrony 6u IPv6 6922 0t0 UDP [::1]:323
ssm-agent 1347 root 17u IPv4 9898 0t0 TCP 172.31.75.85:40920->98.87.174.97:443 (ESTABLISHED)
sshd 1496 root 3u IPv4 6397 0t0 TCP *:22 (LISTEN)
sshd 1496 root 4u IPv6 6399 0t0 TCP *:22 (LISTEN)
sshd 1497 root 4u IPv4 11314 0t0 TCP 172.31.75.85:22->92.191.14.93:65491 (ESTABLISHED)
sshd 1610 ubuntu 4u IPv4 11314 0t0 TCP 172.31.75.85:22->92.191.14.93:65491 (ESTABLISHED)
lua5.4 28046 prosody 7u IPv4 147580 0t0 TCP *:5222 (LISTEN)
lua5.4 28046 prosody 8u IPv6 147581 0t0 TCP *:5222 (LISTEN)
lua5.4 28046 prosody 10u IPv4 147582 0t0 TCP *:5269 (LISTEN)
lua5.4 28046 prosody 11u IPv6 147583 0t0 TCP *:5269 (LISTEN)
lua5.4 28046 prosody 16u IPv4 147591 0t0 TCP 172.31.75.85:5222->92.191.14.93:1620 (ESTABLISHED)
root@im07:/etc/prosody# sudo lsof -i -P -n | grep prosody
lua5.4 28046 prosody 7u IPv4 147580 0t0 TCP *:5222 (LISTEN)
lua5.4 28046 prosody 8u IPv6 147581 0t0 TCP *:5222 (LISTEN)
lua5.4 28046 prosody 10u IPv4 147582 0t0 TCP *:5269 (LISTEN)
lua5.4 28046 prosody 11u IPv6 147583 0t0 TCP *:5269 (LISTEN)
lua5.4 28046 prosody 16u IPv4 147591 0t0 TCP 172.31.75.85:5222->92.191.14.93:1620 (ESTABLISHED)
root@im07:/etc/prosody#
```

Para ver el log y poder comprobar su funcionamiento en tiempo real.

sudo sudo tail -f /var/log/prosody/prosody.log

```
root@im07:/etc/prosody# sudo sudo tail -f /var/log/prosody/prosody.log
Jan 05 15:41:06 c2s573180e17520 debug Received[c2s]: <iq type='get' id='purple713c9d34' xml:lang='en'>
Jan 05 15:41:06 c2s573180e17520 debug Sending[c2s]: <iq type='result' to='admin@im07.institutodh.net/2o ASIR' id='purp
le713c9d34'>
Jan 05 15:42:06 c2s573180e17520 debug Received[c2s]: <iq type='get' id='purple713c9d35' xml:lang='en'>
Jan 05 15:42:06 c2s573180e17520 debug Sending[c2s]: <iq type='result' to='admin@im07.institutodh.net/2o ASIR' id='purp
le713c9d35'>
Jan 05 15:43:06 c2s573180e17520 debug Received[c2s]: <iq type='get' id='purple713c9d36' xml:lang='en'>
Jan 05 15:43:06 c2s573180e17520 debug Sending[c2s]: <iq type='result' to='admin@im07.institutodh.net/2o ASIR' id='purp
le713c9d36'>
Jan 05 15:44:06 c2s573180e17520 debug Received[c2s]: <iq type='get' id='purple713c9d37' xml:lang='en'>
Jan 05 15:44:06 c2s573180e17520 debug Sending[c2s]: <iq type='result' to='admin@im07.institutodh.net/2o ASIR' id='purp
le713c9d37'>
Jan 05 15:45:06 c2s573180e17520 debug Received[c2s]: <iq type='get' id='purple713c9d38' xml:lang='en'>
Jan 05 15:45:06 c2s573180e17520 debug Sending[c2s]: <iq type='result' to='admin@im07.institutodh.net/2o ASIR' id='purp
le713c9d38'>
Jan 05 15:46:06 c2s573180e17520 debug Received[c2s]: <iq type='get' id='purple713c9d39' xml:lang='en'>
Jan 05 15:46:06 c2s573180e17520 debug Sending[c2s]: <iq type='result' to='admin@im07.institutodh.net/2o ASIR' id='purp
le713c9d39'>
```

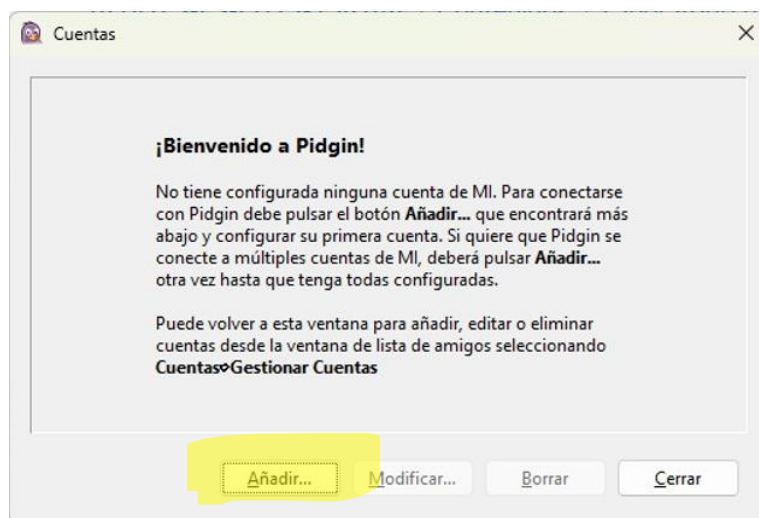
Cliente Pidgin

Una vez que hemos configurado Prosody, vamos a instalar el cliente Pidgin en un equipo cliente. Tenemos versión para Windows, Linux, FreeBSD y macOS⁶.

- **Windows:** Descargar “.exe” y ejecutar.
- **Linux:** apt update y apt install pidgin

Añadir cuenta a Pidgin⁷.

El primer paso es añadir una cuenta del dominio a Pidgin.



⁶ <https://pidgin.im/install/>

⁷ <https://www.youtube.com/watch?v=nD1ujGbQyA0>

Elegimos el protocolo XMPP y luego introducimos las credenciales de alguno de los usuarios que hemos creado antes. También indicar el dominio.

Al final hay una opción de “**Crear esta nueva cuenta en el servidor**”, que por su nombre nos permitiría crear una cuenta directamente sin usar el comando “**prosodyctl**”.

Añadir cuenta

Básica Avanzadas Pasarela

Opciones de conexión

Protocolo: XMPP

Nombre de usuario: admin

Dominio: im07.institutodh.net

Recurso: 2º ASIR

Contraseña: •••••


☒ Recordar contraseña

Opciones de usuario

Apodo local: Administrador

☒ Notificaciones de correo nuevo

☒ Utilizar este icono de amigo para esta cuenta:

 Quitar

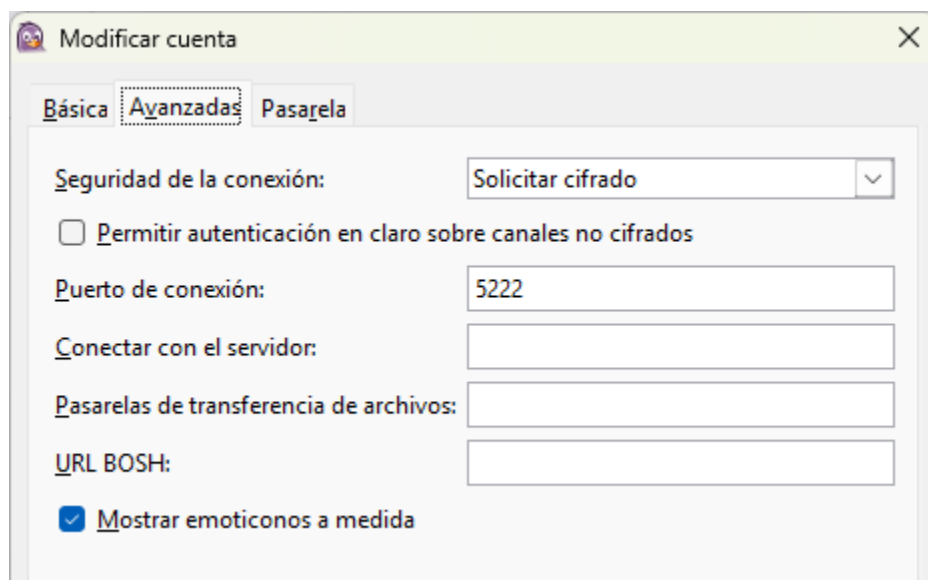
☐ Crear esta nueva cuenta en el servidor

Cancelar Añadir

En algunos manuales⁸ se indica que en “Recursos” hay que poner “*the name of the plaintext port (5222). StartTLS will be used for encryption*”. En nuestro caso no lo hemos puesto y aun así ha funcionado.

⁸ <https://www.ssltrust.com.au/help/setup-guides/xmpp-over-ssl-tls-ubuntu>

En la pestaña de “**Avanzada**” ya nos aparece el puerto especificado. Aquí debemos indicar en “Seguridad de la Conexión” → “**Solicitar cifrado**”, para que use SSL/TSL.



Modificar cuenta

Básica **Avanzadas** Pasarela

Seguridad de la conexión: Solicitar cifrado

☐ Permitir autenticación en claro sobre canales no cifrados

Puerto de conexión: 5222

Conectar con el servidor:

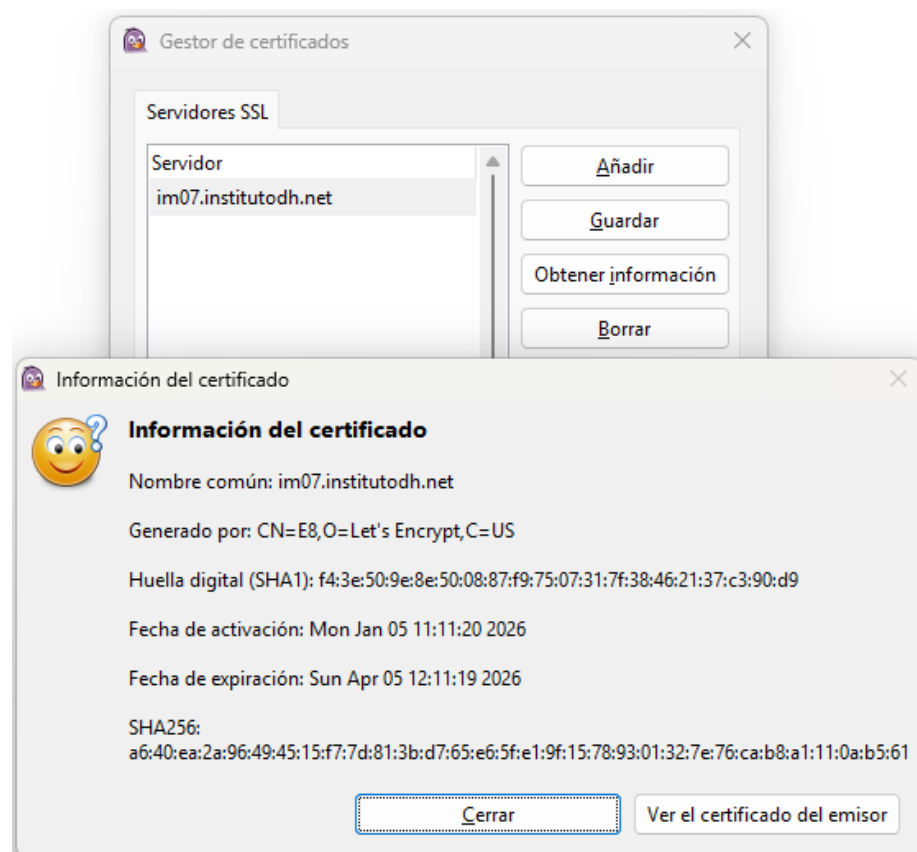
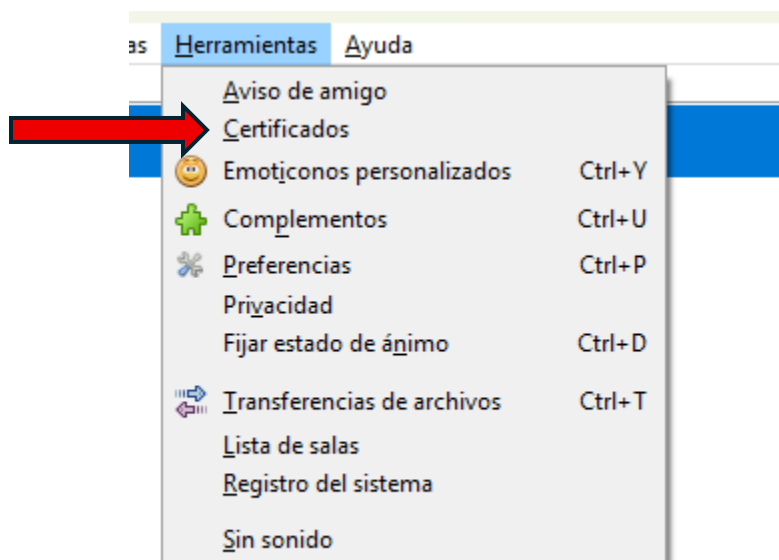
Pasarelas de transferencia de archivos:

URL BOSH:

☒ Mostrar emoticonos a medida

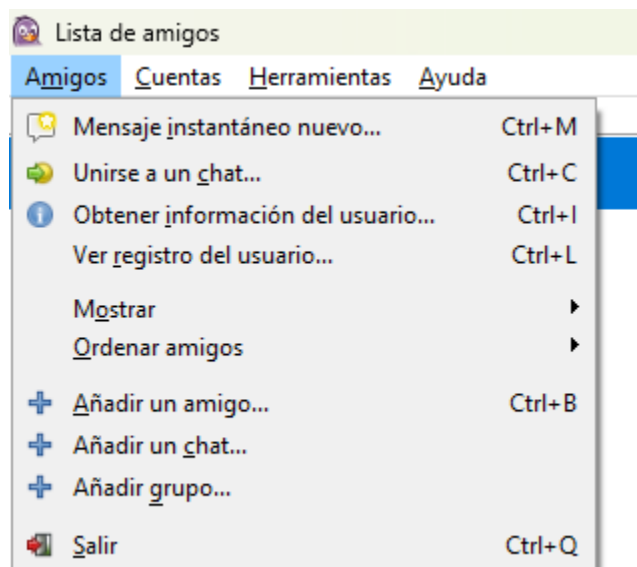
Ver certificados del servidor.

Herramientas → Certificados.

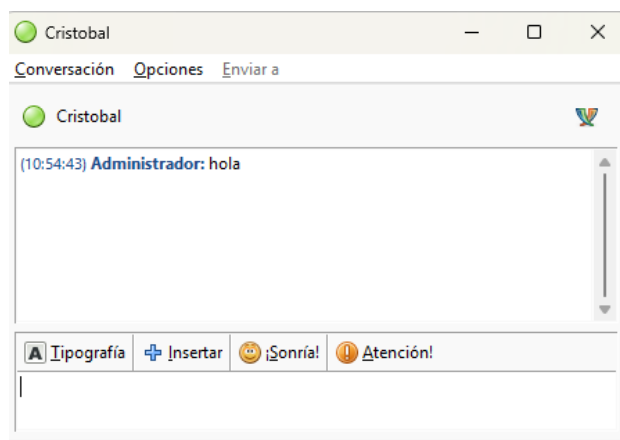
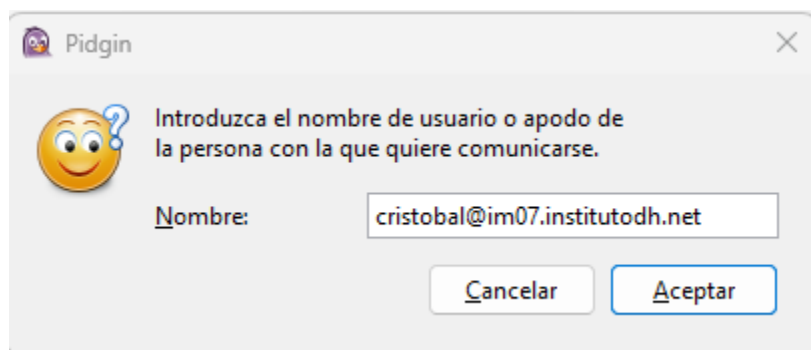


Mensaje instantáneo

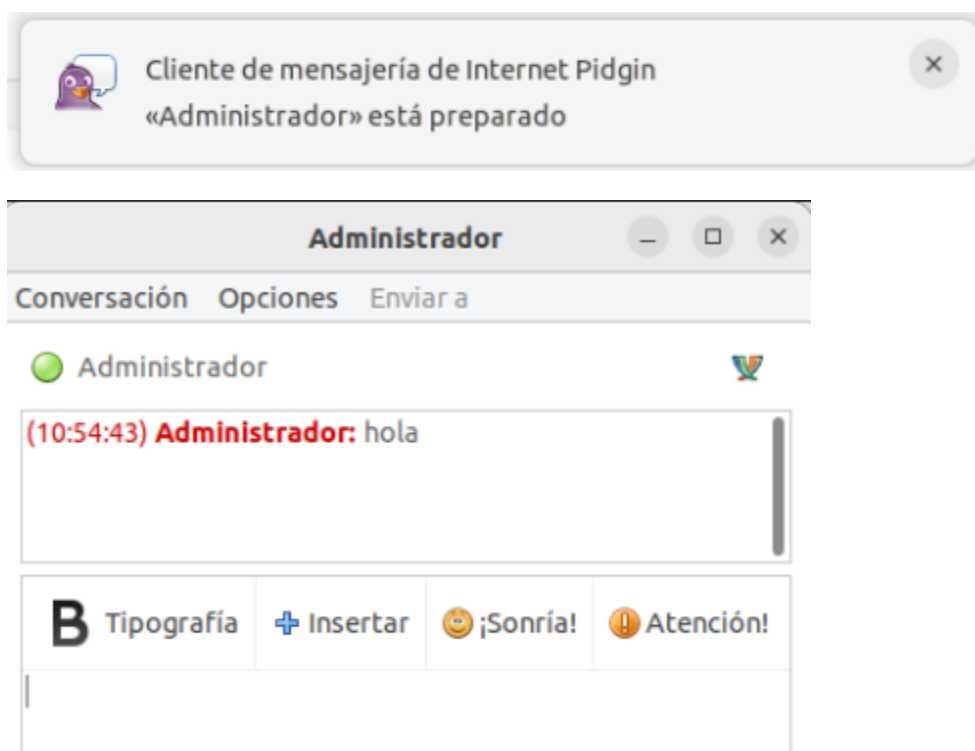
Podemos hablar con otra cuenta, aunque no esté incluida como “amigo”. Para ello usamos “Mensaje instantáneo nuevo...”. → Ctrl+M.



E introducimos la cuenta.

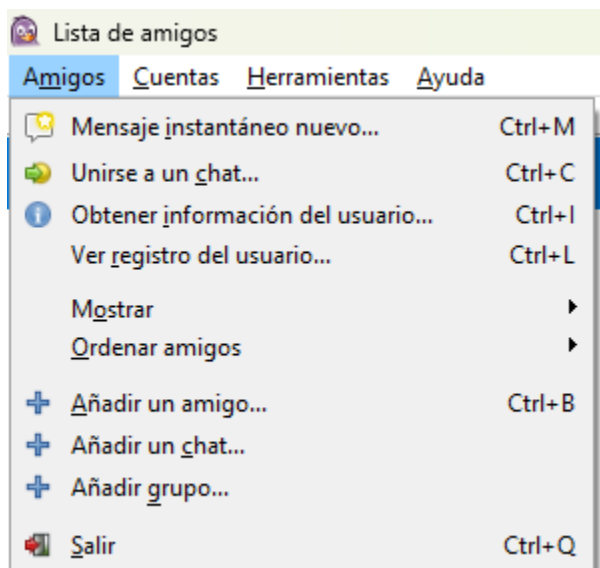


Al otro usuario le debe salir una nueva ventana.

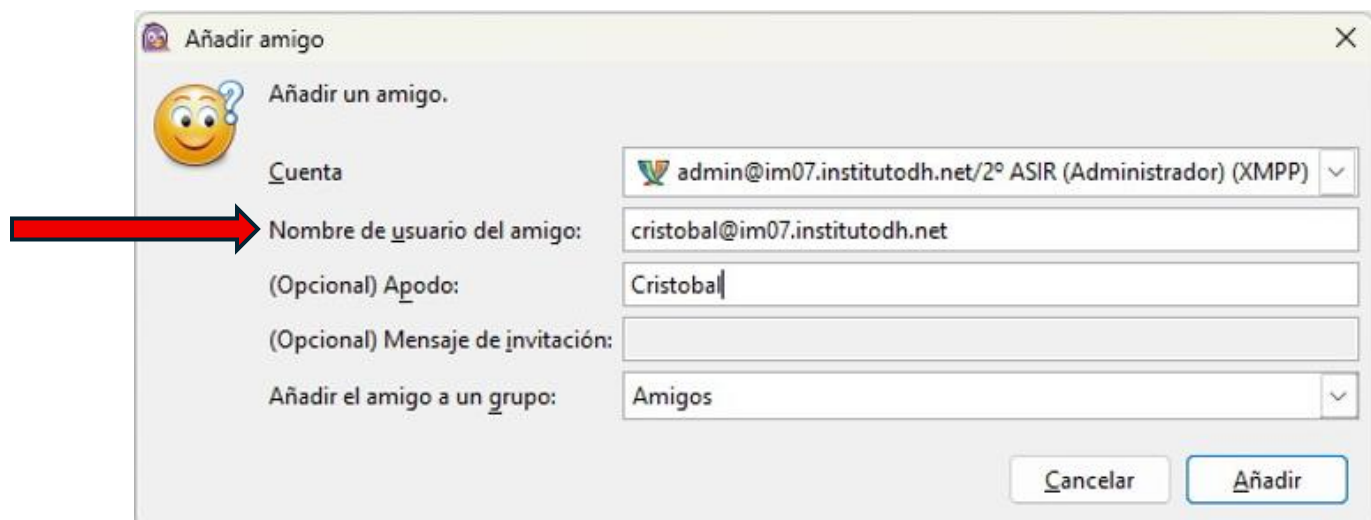


Añadir amigo⁹

Nos vamos a “Añadir un amigo...” → Ctrl+B.



Introducimos las credenciales. En “Nombre de usuario del amigo” es donde tenemos que poner su cuenta.



⁹ <https://www.youtube.com/watch?v=7b1B2sZEHUI>

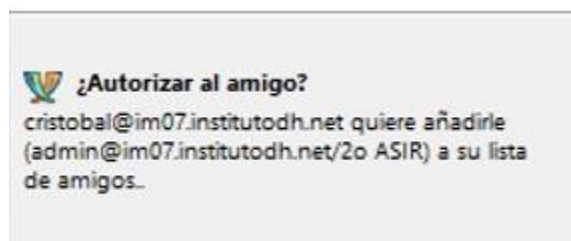
Ahora a nuestro “amigo” le saldrá un mensaje indicando lo que queremos hacer, pidiendo autorización. Puede denegar, autorizar o mandar un mensaje.



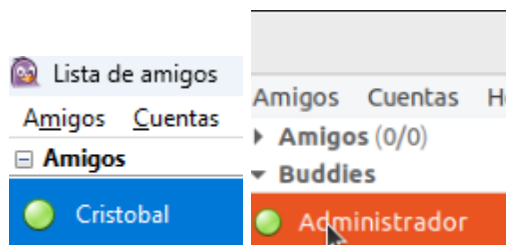
Si le da a autorizar le saldrá una ventana para que incluya al remitente como un amigo.



Y de igual manera, a este le saldrá también la misma notificación que al otro.

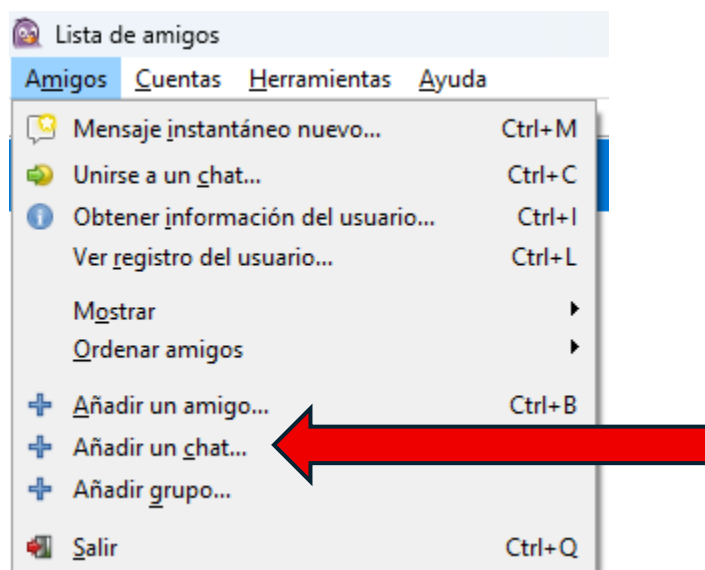


Una vez que el último le haya dado a autorizar, ambos estarán guardados como amigos en la cuenta del otro.



Creación de Salas Abiertas

Amigos → Añadir un chat



Si queremos que sea pública, una buena decisión es no ponerle contraseña. El servidor de chats es “conference.im07.institutodh.net”.

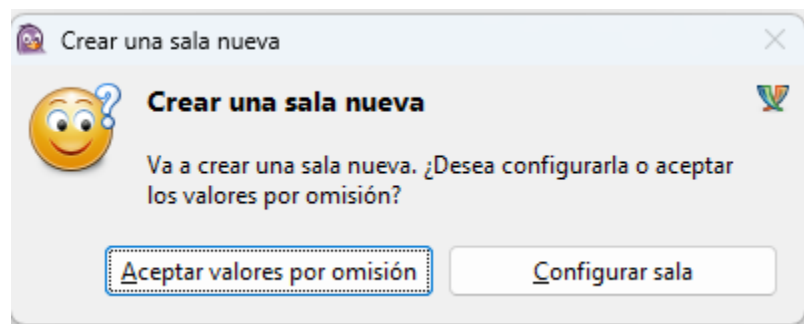
The 'Añadir chat' dialog box contains the following fields and options:

- Cuenta:** admin@im07.institutodh.net/2º ASIR (Administrador) (XMPP)
- Sala:** Sala_Abierta
- Servidor:** conference.im07.institutodh.net
- Manipulador:** admin
- Contraseña:** (empty field)
- Apodo:** (empty field)
- Grupo:** Chats
- ☐ Automatically join when account connects
- ☒ Remain in chat after window is closed

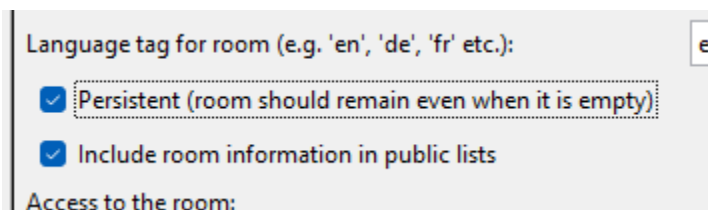
Buttons at the bottom: Lista de salas, Cancelar, Añadir.

La primera vez que le demos al icono del chat nos saldrá una ventana para hacer una configuración más detallada.

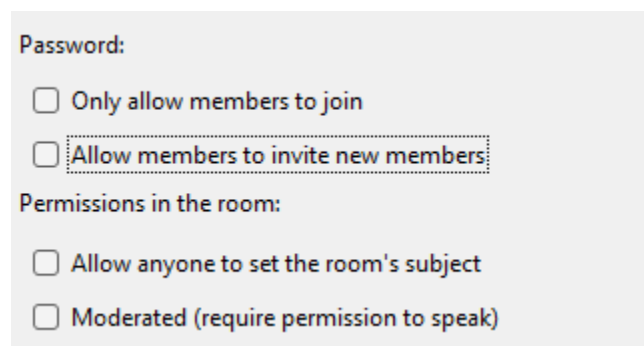
CUIDADO: Una vez que terminas esta configuración no encuentro la manera de volver a encontrarla, así que ten cuidado. Otra advertencia, el primer usuario que le dé abrirá el menú de configuración, así que hazlo rápido.



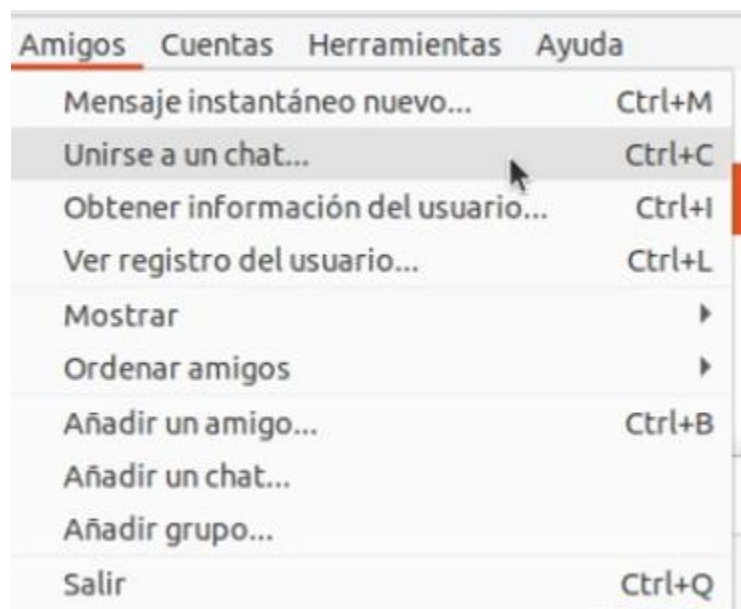
Importante la opción de “**Include room...**”. Esto permitirá listar el chat como veremos más adelante.



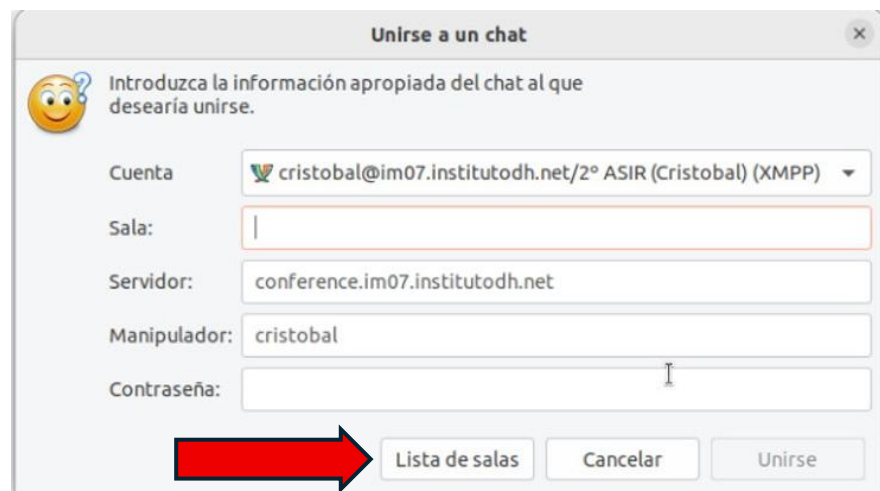
Es un chat abierto, así que puede unirse todo el mundo y no tiene sentido que se permita a los miembros a invitar a nadie, porque no tiene barreras.



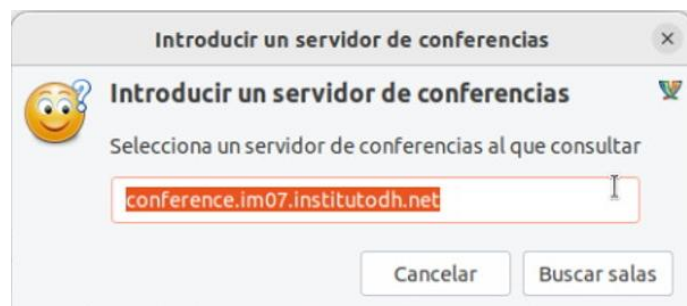
Para unirse al chat



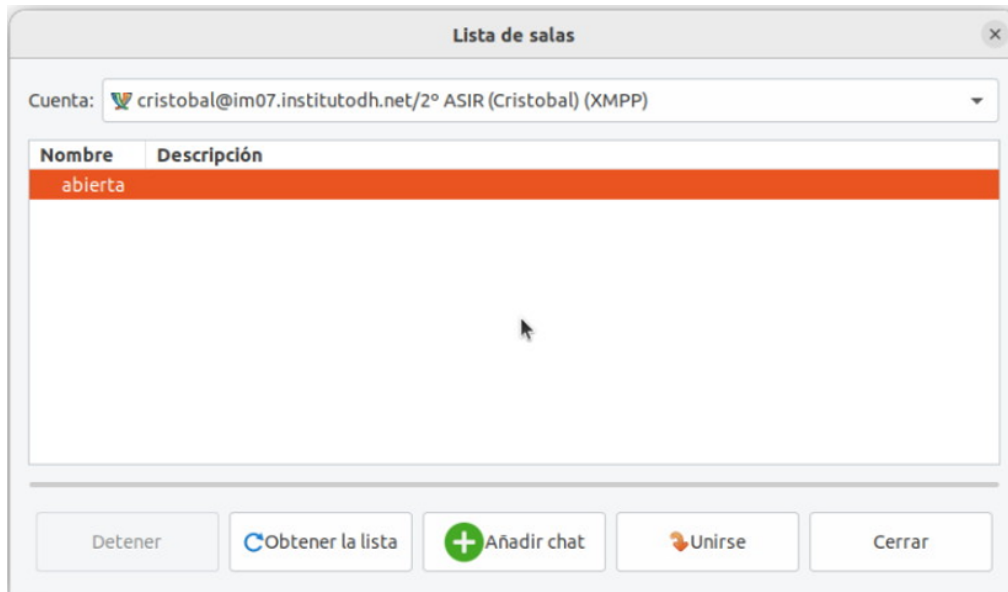
Nos saldrá esta ventana y le damos a “Lista de Salas”



Elegimos el servidor

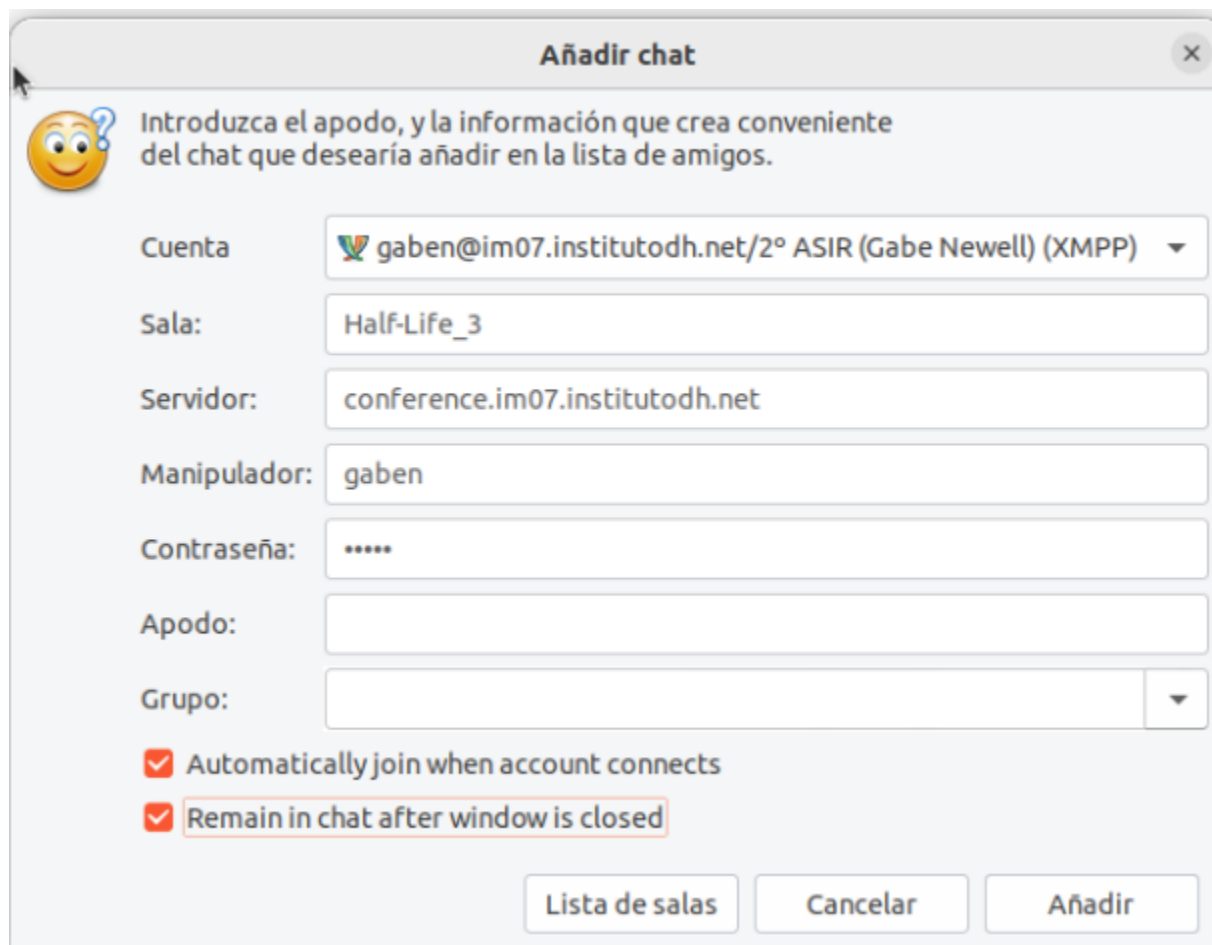


Y ahora nos saldrá la sala. Podremos añadirla y unirnos a ella.



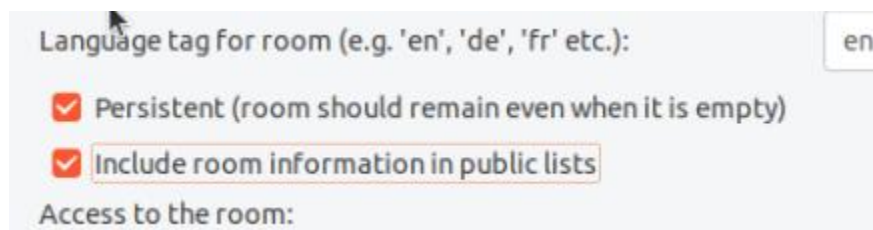
Sala Cerrada

Establecemos una contraseña



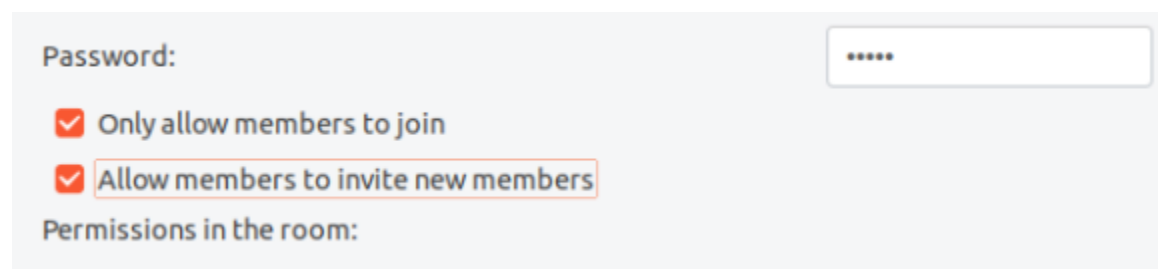
The screenshot shows a dialog box titled "Añadir chat" with a close button (X) in the top right corner. Inside the dialog, there is a yellow smiley face icon with a question mark. The text reads: "Introduzca el apodo, y la información que crea conveniente del chat que desearía añadir en la lista de amigos." Below this, there are several input fields: "Cuenta" (Account) with a dropdown menu showing "gaben@im07.institutodh.net/2º ASIR (Gabe Newell) (XMPP)", "Sala:" (Room) with the text "Half-Life_3", "Servidor:" (Server) with the text "conference.im07.institutodh.net", "Manipulador:" (Handler) with the text "gaben", "Contraseña:" (Password) with masked characters "*****", "Apodo:" (Nickname) with an empty field, and "Grupo:" (Group) with an empty dropdown menu. Below these fields, there are two checked checkboxes: "Automatically join when account connects" and "Remain in chat after window is closed". At the bottom, there are three buttons: "Lista de salas" (List of rooms), "Cancelar" (Cancel), and "Añadir" (Add).

Si queremos que se pueda encontrar en la “Lista de salas”, debemos incluir estas opciones.



The screenshot shows a section titled "Language tag for room (e.g. 'en', 'de', 'fr' etc.):" with a dropdown menu showing "en". Below this, there are two checked checkboxes: "Persistent (room should remain even when it is empty)" and "Include room information in public lists". At the bottom, there is a label "Access to the room:".

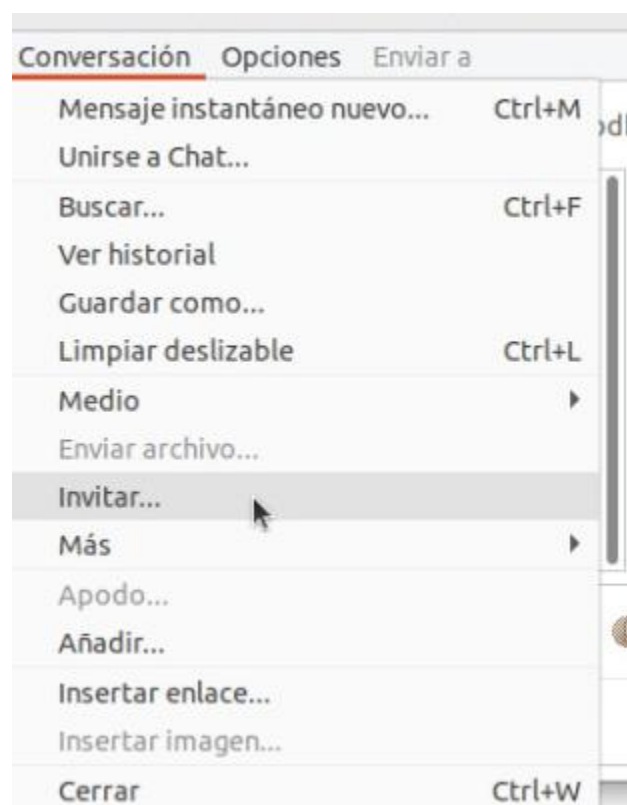
En esta ocasión si hemos establecido una contraseña. Ahora para hacerla de verdad cerrada, podemos usar las opciones de **“Solo permitir unirse a miembros”** y **“Que los miembros puedan invitar nuevos miembros”**. Esto lo haces según sea conveniente.



Formulario de configuración de chat:

- Contraseña: [Campo de texto con caracteres ocultos por puntos]
- ☒ Only allow members to join
- ☒ Allow members to invite new members
- Permissions in the room:

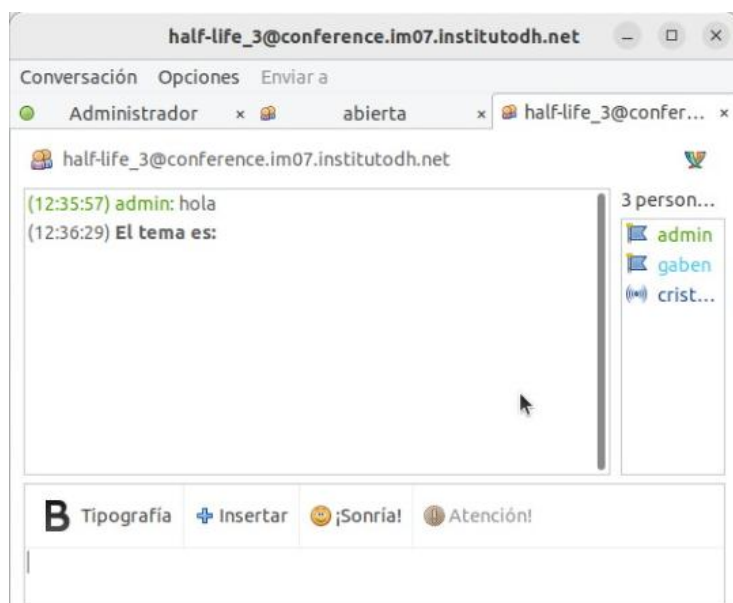
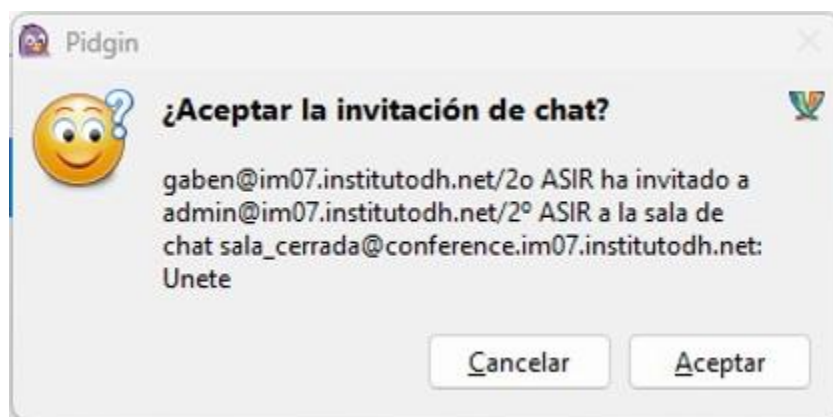
Para incluir a miembros la única opción que encuentro es que el creador del Chat los invite. Para ello, abre el chat → Conversación → Invitar.



Introduce los datos del usuario al que quiere invitar.

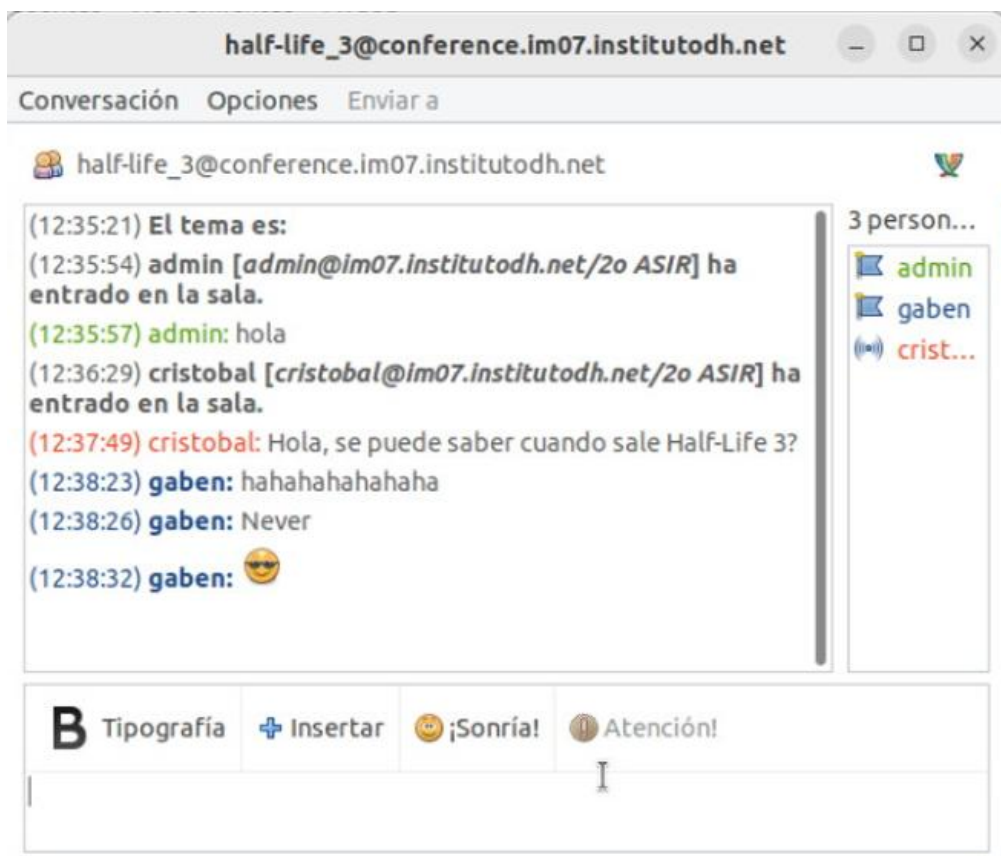


Al otro usuario le saldrá un mensaje. Y una vez que acepte podrá entrar.

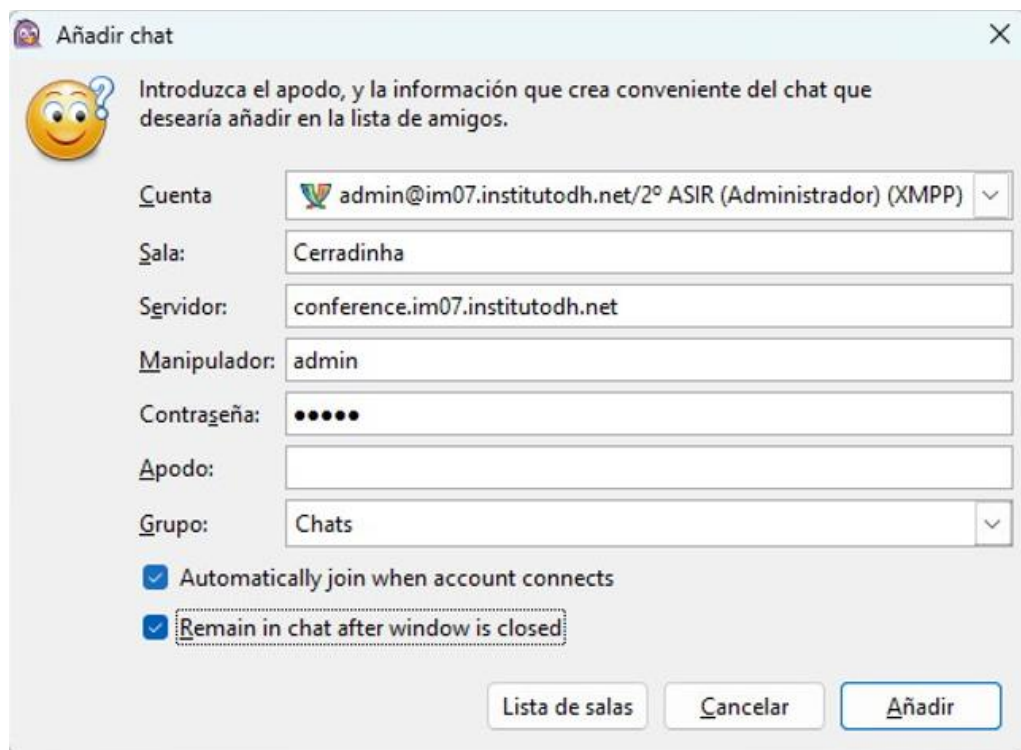


ATENCIÓN: una vez que el usuario invitado cierre la ventana del chat, no podrá volver a entrar sin la invitación del usuario creador.

Como hemos configurado antes que los miembros puedan invitar a otros miembros, podemos hacer que este último invite a otro. Podemos ver como el usuario “admin”, que había sido previamente invitado por “gaben”, ha invitado ahora a “cristobal”.



Sala Cerrada – Solo Contraseña



Añadir chat

Introduzca el apodo, y la información que crea conveniente del chat que desearía añadir en la lista de amigos.

Cuenta: admin@im07.institutodh.net/2º ASIR (Administrador) (XMPP)

Sala: Cerradinha

Servidor: conference.im07.institutodh.net

Manipulador: admin

Contraseña: •••••

Apodo:

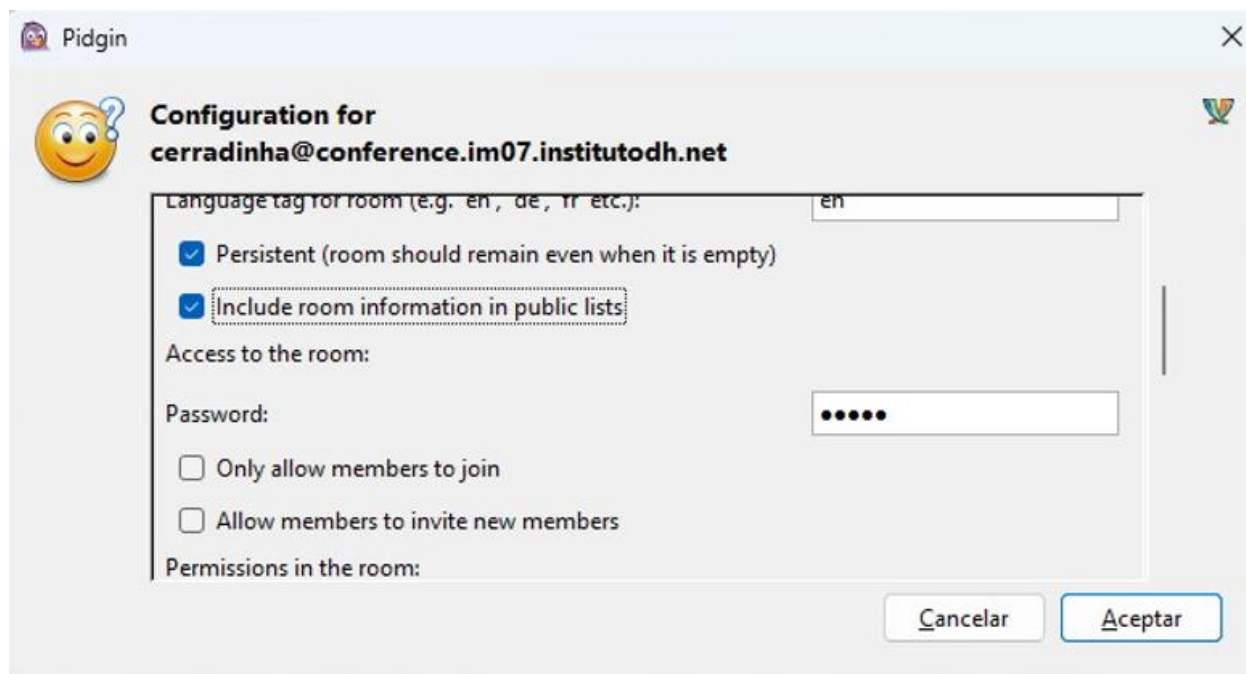
Grupo: Chats

☒ Automatically join when account connects

☒ Remain in chat after window is closed

Lista de salas Cancelar Añadir

En esta ocasión no se usan las opciones de “Solo miembros”.



Pidgin

Configuration for
cerradinha@conference.im07.institutodh.net

Language tag for room (e.g. en, de, fr etc.): en

☒ Persistent (room should remain even when it is empty)

☒ Include room information in public lists

Access to the room:

Password: •••••

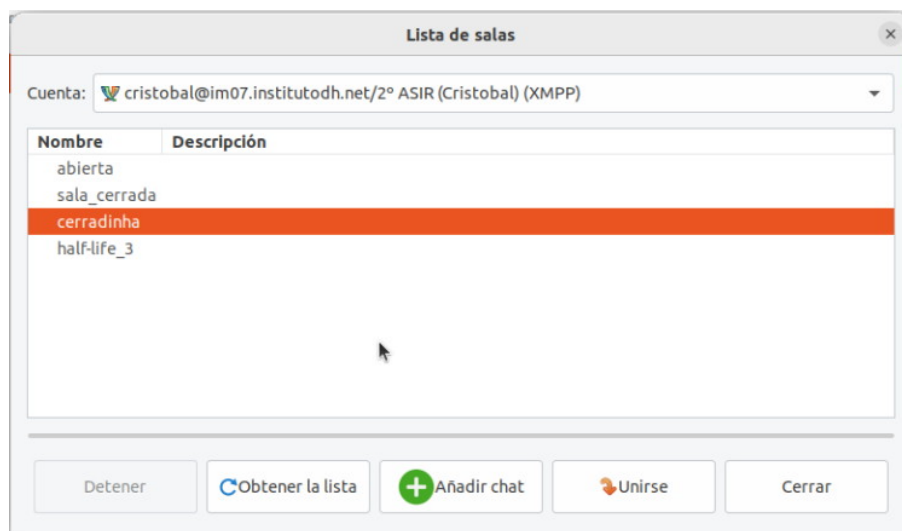
☐ Only allow members to join

☐ Allow members to invite new members

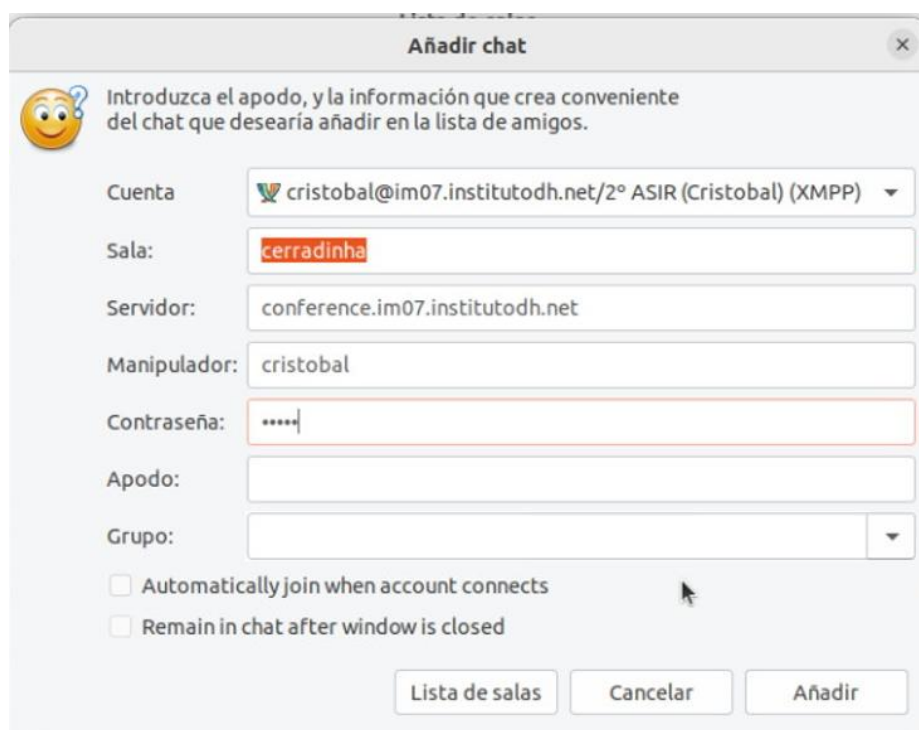
Permissions in the room:

Cancelar Aceptar

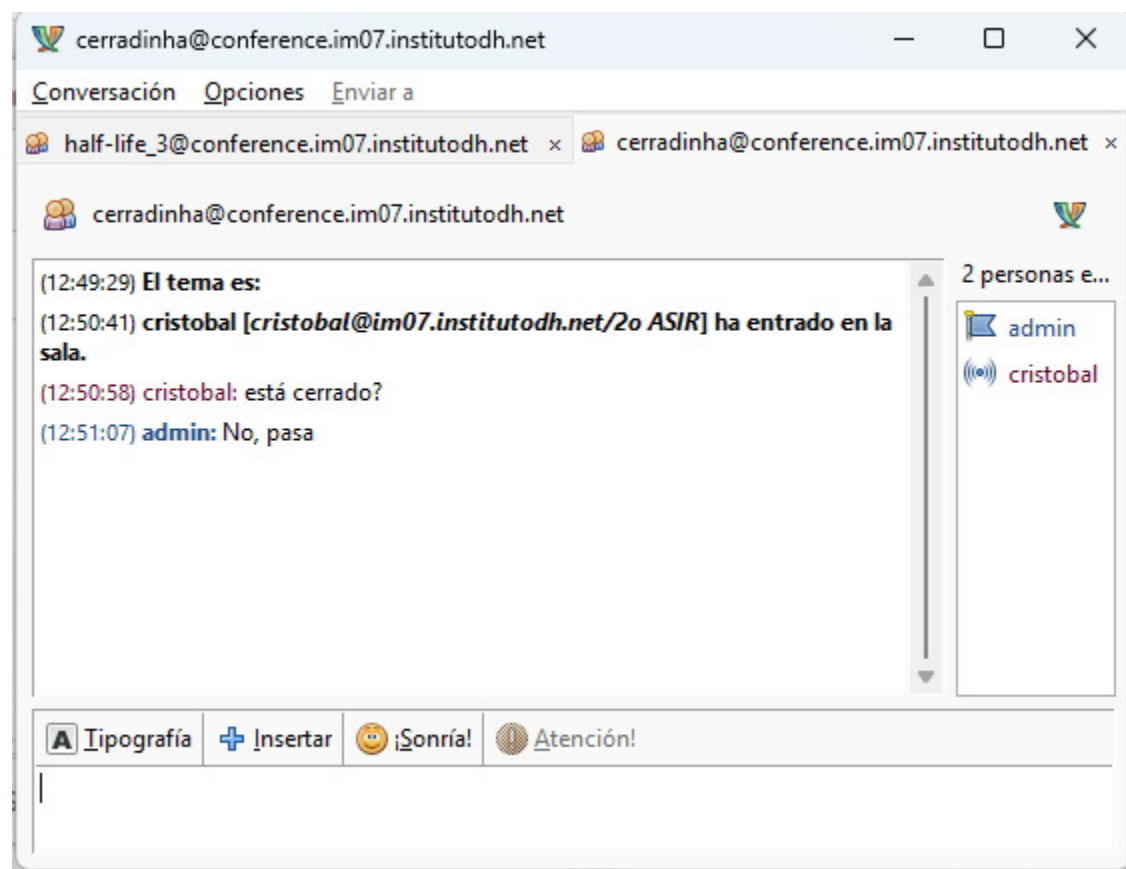
Después los otros usuarios la buscan en la “Lista de salas”. Y le deben dar a “Añadir chat”.



Ahí deben poner la contraseña.

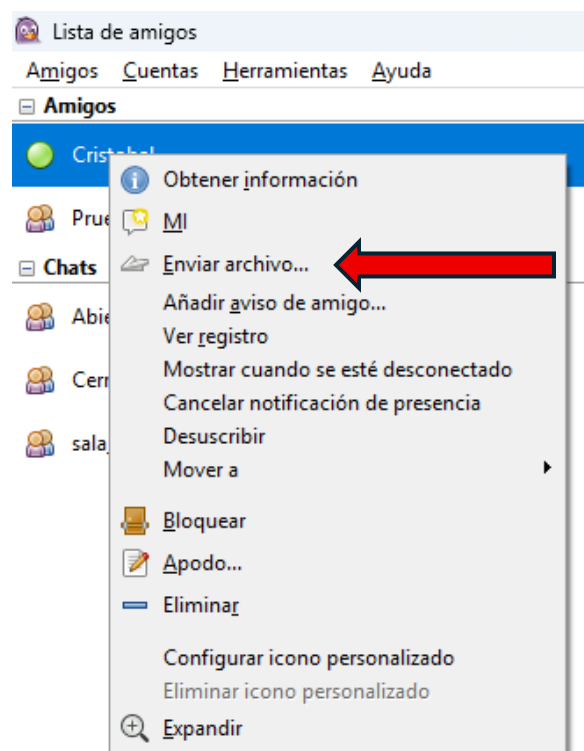


Entonces ya podrán usar el chat.



Enviar archivo

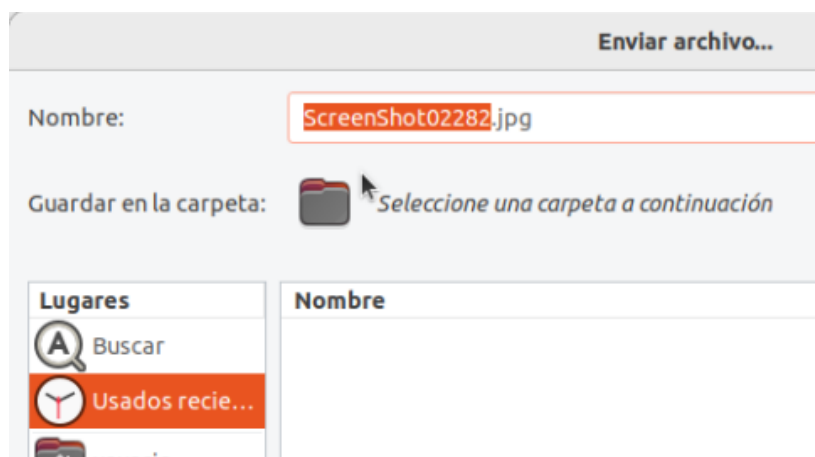
Seleccionamos al usuario al que se lo queremos enviar.



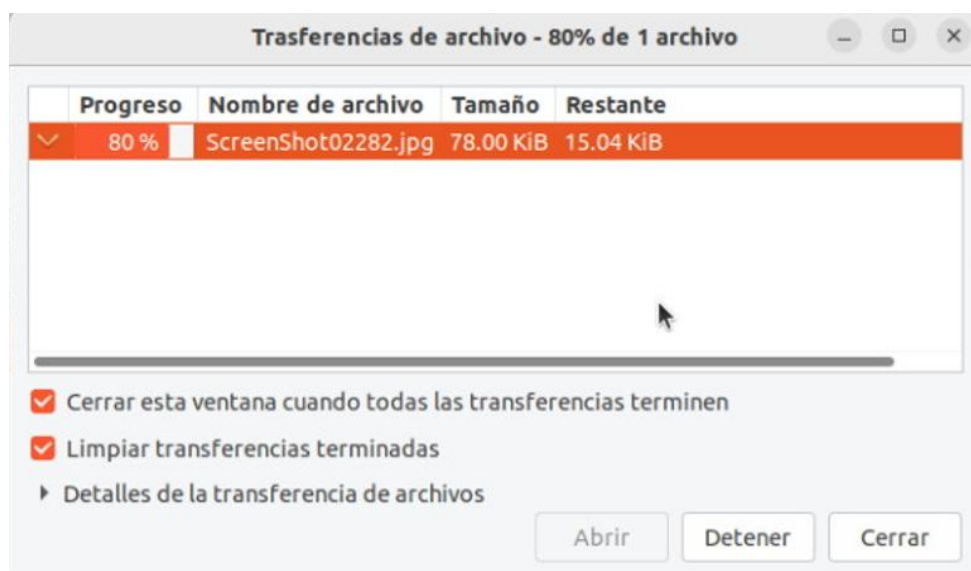
Lo seleccionamos y le damos a enviar. Al otro usuario le saldrá un mensaje como este.



Si le damos a Aceptar, nos saldrá la ventana para decidir el directorio donde guardarlo.



Proceso de descarga.



Instalación de Funcionalidades:

Hay varias maneras de habilitar funcionalidades extras en Prosody. Para ello usamos módulos, los cuales pueden estar ya en el directorio que tenemos definido en:

```
plugin_paths = { "/usr/local/lib/prosody/modules" }
```

O lo podemos descargar desde alguno de los repositorios de la comunidad:

- LuaRocks:

<https://modules.prosody.im/rocks/>

- Alguno de los múltiples recopilatorios en GitHub:

<https://github.com/kcybulski/prosody-modules>

<https://github.com/prosody-modules>

Plugin Installer¹⁰

Para usar el Plugin Installer debemos instalar LuaRocks

- Instalar LuaRocks:

```
sudo apt install luarocks
```

- Comprobar qué versión de Lua usa Prosody

```
prosodyctl about
```

Busca esta línea:

```
# Operating system
Linux 6.14.0-1018-aws

# Lua environment
Lua version:                Lua 5.4

Lua module search paths:
/usr/lib/prosody/?.lua
...

```

Nos indica las dependencias de compilación que debemos instalar:

```
sudo apt install -y liblua5.4-dev build-essential
```

¹⁰ https://prosody.im/doc/plugin_installer

- Crear la carpeta de plugins y dar permisos a Prosody

```
sudo mkdir -p /var/lib/prosody/custom_plugins
```

```
sudo chown -R prosody:prosody /var/lib/prosody/
```

- Comprueba que ahora existen los headers:

```
ls -l /usr/include/lua5.4/lua.h
```

```
root@im07:/home/ubuntu# ls -l /usr/include/lua5.4/lua.h
-rw-r--r-- 1 root root 15949 May  2  2023 /usr/include/lua5.4/lua.h
```

Ahora, en el archivo de configuración “prosody.cfg.lua” podemos especificar la dirección del servidor desde el cual nos vamos a bajar los módulos:

```
plugin_server = https://modules.prosody.im/rocks/
```

```
plugin_server = "https://modules.prosody.im/rocks/"
installer_plugin_path = "/var/lib/prosody/custom_plugins"
```

En el mismo archivo también podemos especificar el directorio donde se instalarán (el que hemos creado antes):

```
installer_plugin_path = "/var/lib/prosody/custom_plugins"
```

```
installer_plugin_path = "/var/lib/prosody/custom_plugins"
```

Se indica que no es necesario indicar este directorio en el “plugin_paths”

“The installer path does **not** need to be added to plugin_paths.”¹¹

Pero hacerlo no perjudica su funcionamiento:

```
plugin_paths = {
    "/usr/lib/prosody/modules",
    "/var/lib/prosody/custom_plugins"
}
plugin_server = "https://modules.prosody.im/rocks/"
```

Guardamos la configuración y reiniciamos prosody.

¹¹ https://prosody.im/doc/plugin_installer

Ahora podremos instalar los módulos sin necesidad de especificar la URL, porque la hemos definido previamente:

```
sudo prosodyctl install mod_admin_web
```

Aun así, si nos da algún problema, siempre podemos especificar al URL de la manera que se muestra abajo

```
sudo prosodyctl install --server=https://modules.prosody.im/rocks/ mod_admin_web
```

- Para listar módulos instalados:

```
sudo prosodyctl list
```

```
root@im07:/home/ubuntu# sudo prosodyctl list

Rocks installed for Lua 5.4 in /var/lib/prosody/custom_plugins
-----

mod_admin_web
  86-1 (installed) - /var/lib/prosody/custom_plugins/lib/luarocks/rocks-5.4
```

- Comprobar que módulos pueden ser actualizados (para actualizar ejecuta de nuevo el comando de instalar).

```
sudo prosodyctl list --outdated
```

```
root@im07:/home/ubuntu# sudo prosodyctl list --outdated

Outdated rocks:
-----
```

En nuestro caso no hay ninguno desactualizado (los módulos no se actualizan desde hace 7-10 años, así que con esto no habrá problemas).

- Para desinstalar módulos:

```
sudo prosodyctl remove mod_example_module
```

ATENCIÓN: “Remember to unload the module and disable it by e.g. removing it from [modules_enabled](#), or Prosody will complain about not finding it the next time it starts.”

Nuevos Módulos y Componentes Instalados:

Ahora vamos a mostrar algunas nuevas funcionalidades añadidas:

`mod_admin_web`¹²

Provee de una interfaz web que permite gestionar ciertos aspectos del servidor Prosody.

Este módulo no viene por defecto con Prosody, así que el primer paso es descargárnoslo:

`sudo prosodyctl install --server=https://modules.prosody.im/rocks/ mod_admin_web`

Si en vez de la versión de LuaRocks nos lo queremos descargar de otro sitio, solo tenemos que cambiar la URL.

ATENCIÓN: En las instrucciones oficiales se menciona la ejecución de un script “`get_deps.sh`” que está en los archivos descargados; pero en la versión de LuaRocks no aparece. En las otras versiones de GitHub si existe este script.

Una vez descargado los ficheros, debemos “activar” el módulo en el archivo “`prosody.cfg.lua`”, en la sección de “`modules_enabled`”:

`"admin_web";`

```
"register"; -- Allow
-- Admin interfaces
"admin_adhoc"; -- AL
"admin_web"; -- ESTE
"bosh"; -- https://p
"http"; -- https://p
"admin_shell"; -- ht
```

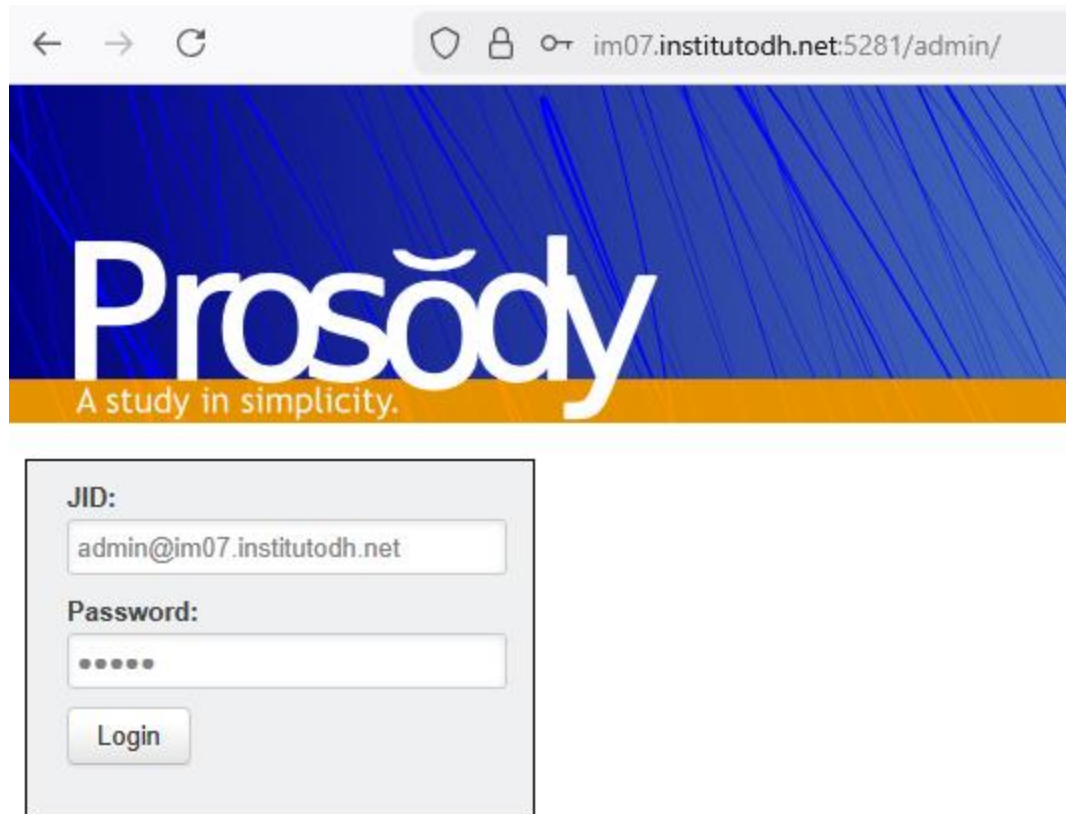
Reiniciamos el servicio Prosody y podremos acceder desde el navegador de un cliente desde esta URL:

“By default the interface will then be reachable under <http://example.com:5280/admin> , or <https://example.com:5281/admin>.”

En nuestro caso:

<https://im07.institutodh.net:5281/admin>

¹² https://modules.prosody.im/mod_admin_web



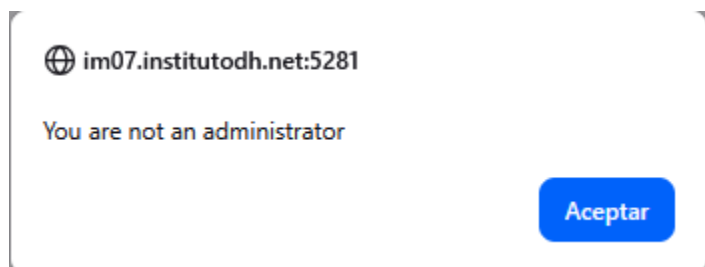
The screenshot shows a web browser window with the address bar displaying `im07.institutodh.net:5281/admin/`. The page features a blue header with the word "Prosody" in large white letters and the tagline "A study in simplicity." below it. Below the header is a login form with the following fields and elements:

- JID:** A text input field containing `admin@im07.institutodh.net`.
- Password:** A password input field with masked characters (dots).
- Login:** A button to submit the login credentials.

Solo nos permitirá entrar con las credenciales del administrador, el cual ha sido especificado en pasos anteriores en el archivo "prosody.cfg.lua":

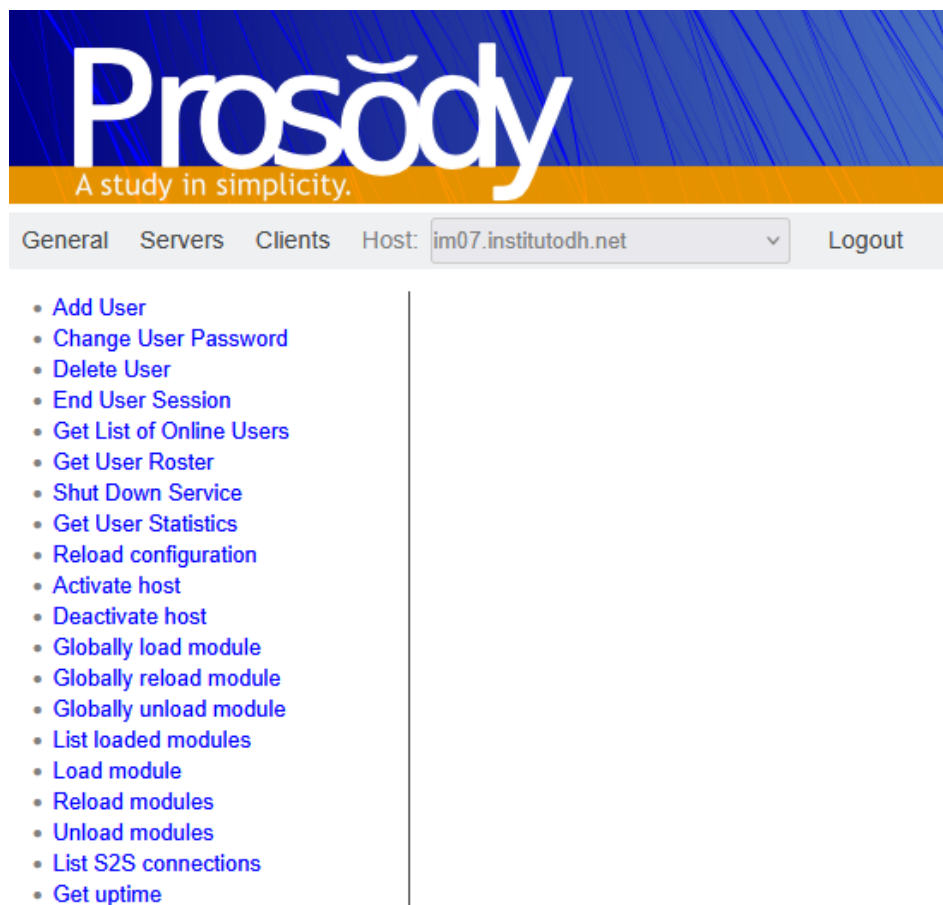
```
GNU nano 7.2
admins = { "admin@im07.institutodh.net" }
plugin_paths = {
```

En caso contrario:



The screenshot shows a web browser window with the address bar displaying `im07.institutodh.net:5281`. The page displays the message "You are not an administrator" and a blue button labeled "Aceptar".

Una vez dentro tendremos el siguiente listado de funcionalidades:



tombstones

“Este módulo deja una **“lápida” (tombstone)** de la cuenta de un usuario después de que sea eliminada. Esto hace imposible que otro usuario registre una cuenta y **suplante al usuario original**, incluso después de que la cuenta original haya sido borrada.”¹³

Este viene incluido en /usr/lib/prosody/modules/

```
root@im07:/home/ubuntu# ls -l /usr/lib/prosody/modules/ | grep tombstones
-rw-r--r-- 1 root root 3897 Sep 5 2023 mod_tombstones.lua
```

Para activarlo solo hay que indicarlo en la sección de módulos habilitados:

```
-- Módulos extras
    "tombstones"; -- Prevent registration of deleted accounts
```

Y también, el periodo de tiempo por el que estará bloqueado.

user_tombstone_expiry = <tiempo>

Valor	Significado
nil (por defecto)	Nunca caducan
"1d"	1 día
"7d"	7 días
"30d"	30 días
"6m"	6 meses
"1y"	1 año

```
-- Configuración de tombstones. Periodo de tiempo por el que estará bloqueado
user_tombstone_expiry = "1y"
```

¹³ <https://comfy.guide/server/prosody/#tombstones>

mod_http_file_share¹⁴

“Permite a los usuarios compartir archivos, como imágenes, vídeos y clips de audio. Funciona con chats de grupo, múltiples clientes y destinatarios que estén desconectados.”

Es un módulo que ya viene incluido. Para activarlo tenemos que incluir en “prosody.cfg.lua”:

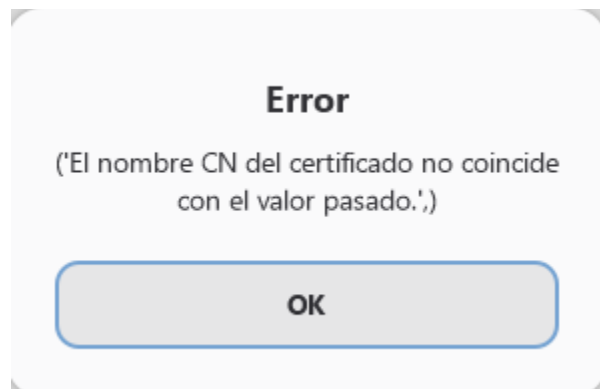
Component "upload.example.org" "http_file_share"

En nuestro caso:

```
-- Configuración para subir imagenes???  
  
Component "upload.im07.institutodh.net" "http_file_share"
```

Es necesario hacer la configuración en Godaddy para que el DNS entienda la dirección de “upload.im07.institutodh.net”. Esta parte la hace Diego (le ha puesto un alias para que use la misma IP que “im07.institutodh.net”).

El cliente Pidgin parece que no tiene esta funcionalidad implementada, así que vamos a usar Gajim para esta parte, porque lo trae de serie. Si intentamos mandar una imagen por el chat en estos momentos, nos saldrá este error:



Debemos generar un certificado para “upload.im07.institutodh.net”.

¹⁴ https://prosody.im/doc/modules/mod_http_file_share

Paso 1 — Reemitir certificado con SAN múltiple

Vuelves a usar Certbot, pero añadiendo el subdominio:

```
sudo certbot certonly --standalone \
```

```
-d im07.institutodh.net \
```

```
-d upload.im07.institutodh.net
```

```
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/im07.institutodh.net/fullchain.pem
Key is saved at: /etc/letsencrypt/live/im07.institutodh.net/privkey.pem
This certificate expires on 2026-04-17.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.
```

Recuerda que el Puerto 80 debe estar abierto.

Paso 2 — Reimportar en Prosody

```
sudo prosodyctl --root cert import /etc/letsencrypt/live
```

```
ls -l /etc/prosody/certs/
```

```
root@im07:/home/ubuntu# ls -l /etc/prosody/certs/
total 16
-rw-r----- 1 prosody prosody 2925 Jan 17 11:18 im07.institutodh.net.crt
-rw-r----- 1 prosody prosody 2880 Jan 5 15:22 im07.institutodh.net.crt.bkp~2026-01-17T11:17:44
-r----- 1 prosody prosody 241 Jan 17 11:18 im07.institutodh.net.key
-r----- 1 prosody prosody 241 Jan 5 15:22 im07.institutodh.net.key.bkp~2026-01-17T11:17:44
root@im07:/home/ubuntu# sudo systemctl restart prosody
```

Paso 3 — Reiniciar Prosody

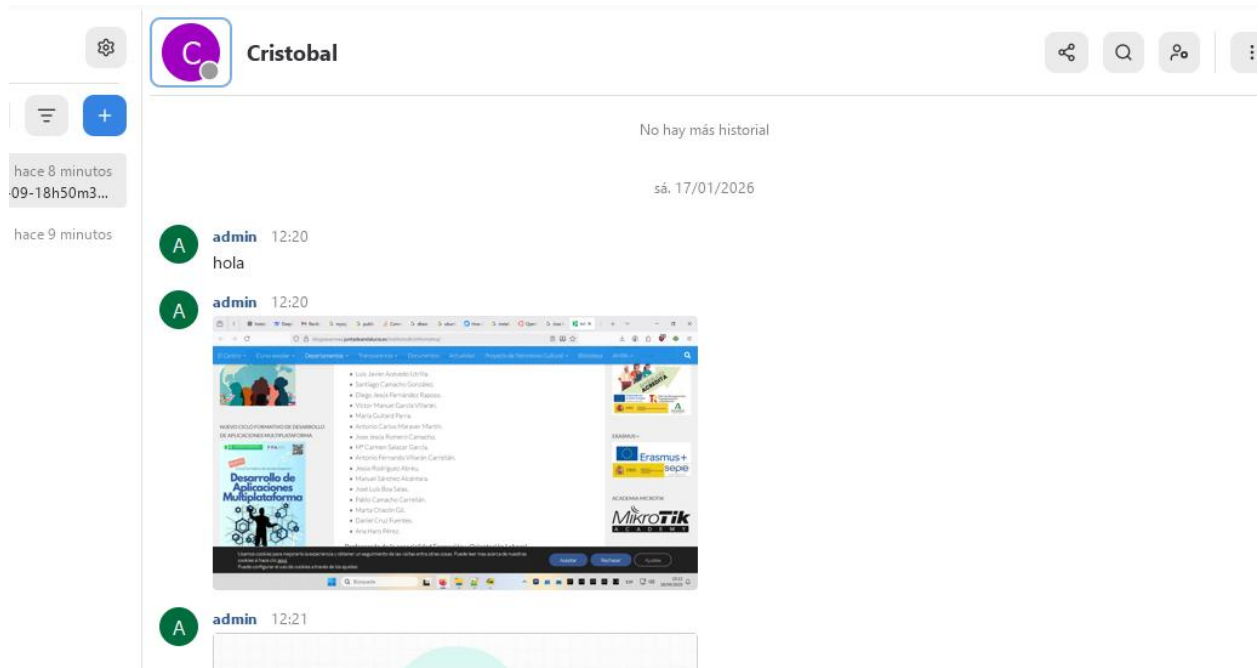
```
sudo systemctl restart prosody
```

```
sudo systemctl status prosody
```

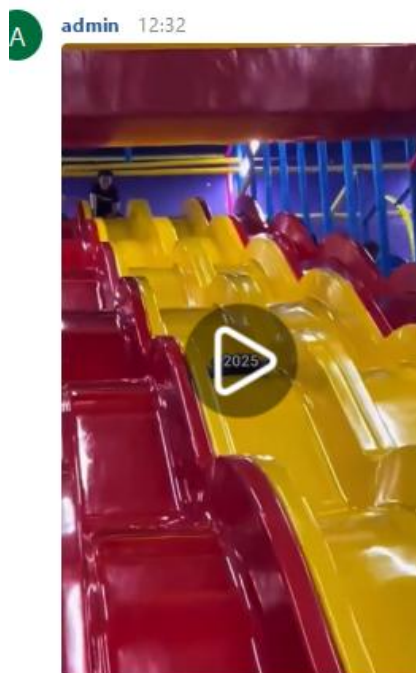
```
root@im07:/home/ubuntu# sudo systemctl status prosody
● prosody.service - Prosody XMPP Server
   Loaded: loaded (/usr/lib/systemd/system/prosody.service; enabled; preset: enabled)
   Active: active (running) since Sat 2026-01-17 11:19:27 UTC; 8s ago
     Docs: https://prosody.im/doc
   Main PID: 81309 (lua5.4)
    Tasks: 1 (limit: 2213)
   Memory: 6.6M (peak: 6.9M)
      CPU: 232ms
   CGroup: /system.slice/prosody.service
           └─81309 lua5.4 /usr/bin/prosody -F

Jan 17 11:19:27 im07 systemd[1]: Started prosody.service - Prosody XMPP Server.
root@im07:/home/ubuntu#
```

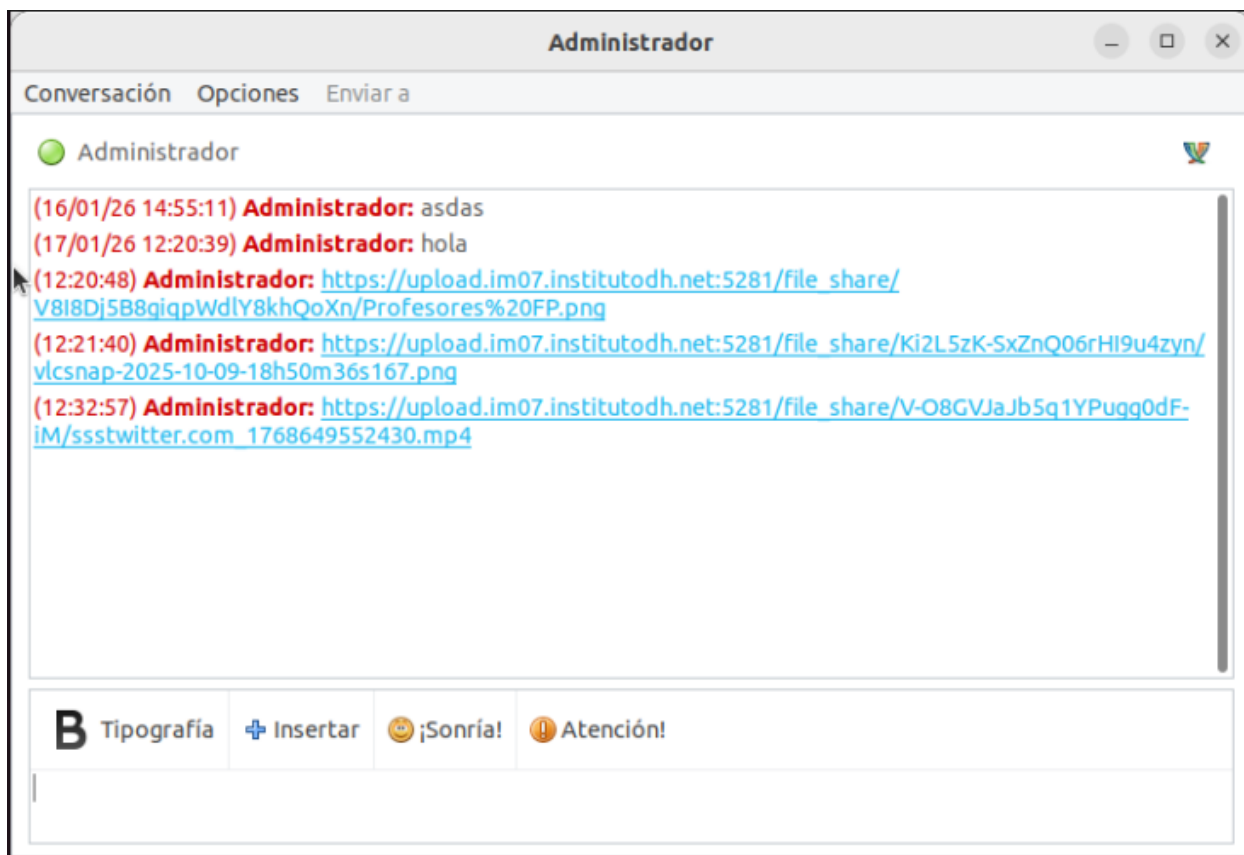
Ahora podemos pasar imágenes por el chat:



Videos (aunque no se ejecutan en el cliente XMPP, necesita un reproductor del sistema).



El usuario receptor no podrá verlos directamente en el cliente Pidgin:



Pero podrá descargarlos:



Probamos con Gajim:

Método 1 de instalación de Gajim:

1. Update your package lists

```
sudo apt update
```

2. Install the main Gajim package

```
sudo apt install gajim
```

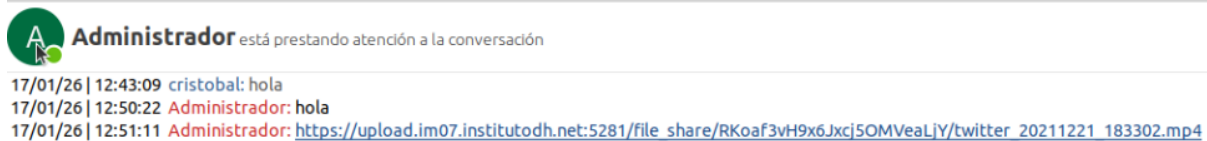
3. (Optional) Install popular plugins like OMEMO for encryption

```
sudo apt install gajim-omemo
```

4. (Optional) Install other plugins as needed (e.g., spell check)

```
sudo apt install gajim-spellcheck
```

Tampoco se puede:



De igual manera que en Pidgin, hay que descargarlos.

Método 2 de Instalación de Gajim:

- Primero hay que instalar flatpak y añadir el repositorio de flathub¹⁵:

```
sudo apt install flatpak
```

```
sudo apt install gnome-software-plugin-flatpak
```

```
flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo
```

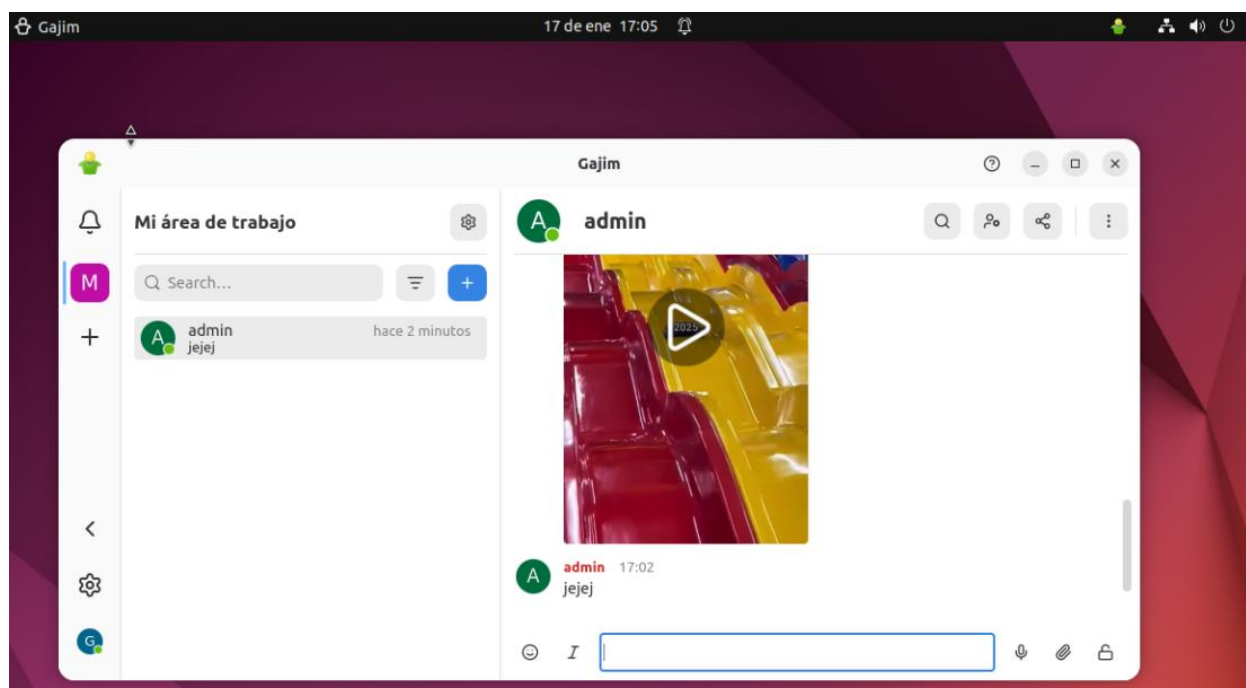
- Después, instalar Gajim¹⁶:

```
flatpak install flathub org.gajim.Gajim
```

¹⁵ <https://flathub.org/en/setup/Ubuntu>

¹⁶ <https://flathub.org/en/apps/org.gajim.Gajim>

Ahora si se pueden ver las imágenes y las “preview” de los videos.



ATENCIÓN: Esta versión, más nueva, es bastante inestable. Se desaconseja el uso de Gajim en clientes Linux.

En este módulo también podemos:

- Especificar el tamaño máximo de los archivos:

`http_file_share_size_limit = 16*1024*1024 -- 16 MiB`

```
Component "upload.im07.institutodh.net" "http_file_share"
http_file_share_size_limit = 1024 -- 1 KB
```



- Cuota diaria por usuario (se limita no por número de archivos, sino la suma de su tamaño)

`http_file_share_daily_quota = 100*1024*1024 -- 100 MiB per day per user`

- Cuota diaria global: limita el máximo diario de todos los usuarios

`http_file_share_global_quota = 1024*1024*1024 -- 1 GiB total`

- Limitar acceso: la configuración de fábrica permite que cualquier usuario pueda usar este módulo. Para limitarlo se usa una configuración parecida a esta:

`http_file_share_access = {`

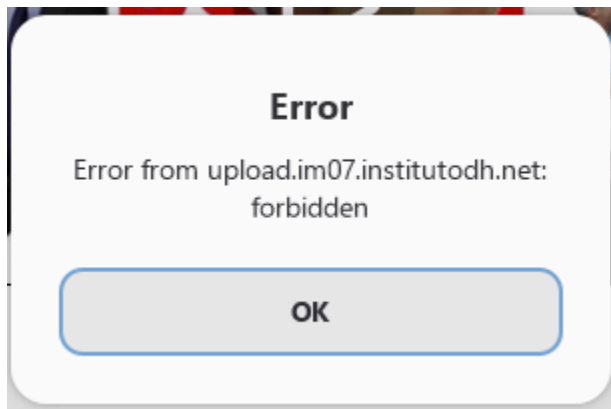
`"filesharingenthusiast@example.net", -- this specific user`

`"example.org", -- anyone with a @example.org address`

`}`

```
http_file_share_access = {
"crisobal@im07.institutodh.net", -- this specific user
}
```

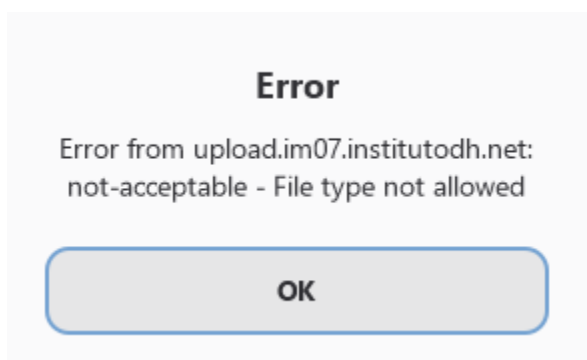
Indicamos quien **SI** puede enviar archivos. Aquellos que no puedan, recibirán un mensaje de “FORBIDDEN” → Prohibido.



- Restringir el tipo de archivos: En el ejemplo solo permite enviar imágenes.

```
http_file_share_allowed_file_types = { "image/*" }
```

```
http_file_share_allowed_file_types = { "image/*" }
```



mod_mam¹⁷

“Por defecto, Prosody solo enviará los mensajes recibidos al **primer cliente disponible**. Esto significa que, si tienes el cliente de escritorio apagado y tu teléfono móvil recibe un mensaje, ese mensaje **no estará disponible para el cliente de escritorio** cuando lo inicies.

Aunque este comportamiento puede ser preferible en algunos casos, puedes habilitar el módulo **mam (Message Archive Management)** para que el servidor **almacene los mensajes y los sincronice con todos los clientes.**”

```
-- Módulos extras
    "tombstones"; -- Prevent registration of deleted accounts
    "mam"; -- Store recent messages to allow multi-device synchronization
}
```

Además:

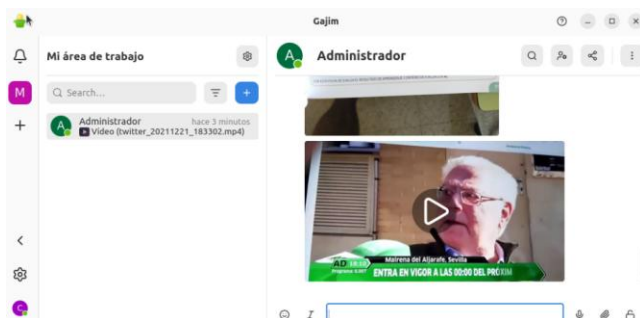
```
-- Configuración mod_mam para sincronización multiples dispositivos

-- Guardar todos los mensajes
default_archive_policy = true

-- Tiempo que se conservan los mensajes
archive_expires_after = "1y"

-- Límite de mensajes por consulta
max_archive_query_results = 50
```

Es difícil de mostrar su funcionamiento en un documento. El usuario había usado su chat en un equipo diferente y cuando volvió abrir sesión en su equipo habitual, todos los cambios habían sido guardados.



¹⁷ <https://comfy.guide/server/prosody/#message-archive-management-chat-history>

mod_checkcerts¹⁸

“Este módulo comprueba periódicamente tu certificado para ver si **está a punto de caducar**. El tiempo restante antes de la expiración se muestra en los registros. Aproximadamente **una semana antes de que caduque un certificado**, se enviarán mensajes de recordatorio a los administradores.”

Nos lo bajamos:

```
sudo prosodyctl install --server=https://modules.prosody.im/rocks/ mod_checkcerts
```

```
root@im07:/home/ubuntu# sudo prosodyctl install --server=https://modules.prosody.im/rocks/ mod_checkcerts
Installing mod_checkcerts in /var/lib/prosody/custom_plugins
Installing https://modules.prosody.im/rocks/mod_checkcerts-16-1.src.rock

mod_checkcerts 16-1 is now installed in /var/lib/prosody/custom_plugins (license: MIT)

Done.
If you installed a prosody plugin, don't forget to add its name under the 'modules_enabled' section inside your configuration file.
Depending on the module, there might be further configuration steps required.

More info about:
  modules_enabled: https://prosody.im/doc/modules_enabled
  mod_checkcerts: https://modules.prosody.im/mod_checkcerts.html
root@im07:/home/ubuntu#
```

Para configurarlo solo hay que ponerlo en “modules_enabled”:

"checkcerts"

```
-- Módulos extras
"tombstones"; -- Prevent registration of deleted accounts
"mam"; -- Store recent messages to allow multi-device synchronization
"checkcerts"; -- https://modules.prosody.im/mod_checkcerts
```

Podemos configurar con cuanto tiempo de antelación queremos que nos avisen:

checkcerts_notify = 7 -- (in days)

```
-- Configuración de checkcert
checkcerts_notify = 7 -- ( in days )
```

¹⁸ https://modules.prosody.im/mod_checkcerts

Desde aquí tendremos acceso a todos los comandos de administración de Prosody.

```
prosody> help
| Commands are divided into multiple sections. For help on a particular section,
| type: help SECTION (for example, 'help c2s'). Sections are:
|
| Section | Description
| c2s      | Commands to manage local client-to-server sessions
| s2s      | Commands to manage sessions between this server and others
| http     | Commands to inspect HTTP services
| module   | Commands to load/reload/unload modules/plugins
| host     | Commands to activate, deactivate and list virtual hosts
| user     | Commands to create and delete users, and change their passwords
| roles    | Show information about user roles
| muc      | Commands to create, list and manage chat rooms
| stats    | Commands to show internal statistics
| server   | Uptime, version, shutting down, etc.
| port     | Commands to manage ports the server is listening on
| dns      | Commands to manage and inspect the internal DNS resolver
| xmpp     | Commands for sending XMPP stanzas
| debug    | Commands for debugging the server
| config   | Reloading the configuration, etc.
| columns  | Information about customizing session listings
| console  | Help regarding the console itself
|
prosody> help user
| user:create(jid, password, roles) - Create the specified user account
| user:password(jid, password) - Set the password for the specified user account
| user:roles(jid, host) - Show current roles for an user
| user:setroles(jid, host, roles) - Set roles for an user (see 'help roles')
| user:delete(jid) - Permanently remove the specified user account
| user:list(hostname, pattern) - List users on the specified host, optionally filtering with a pattern
|
prosody> |
```

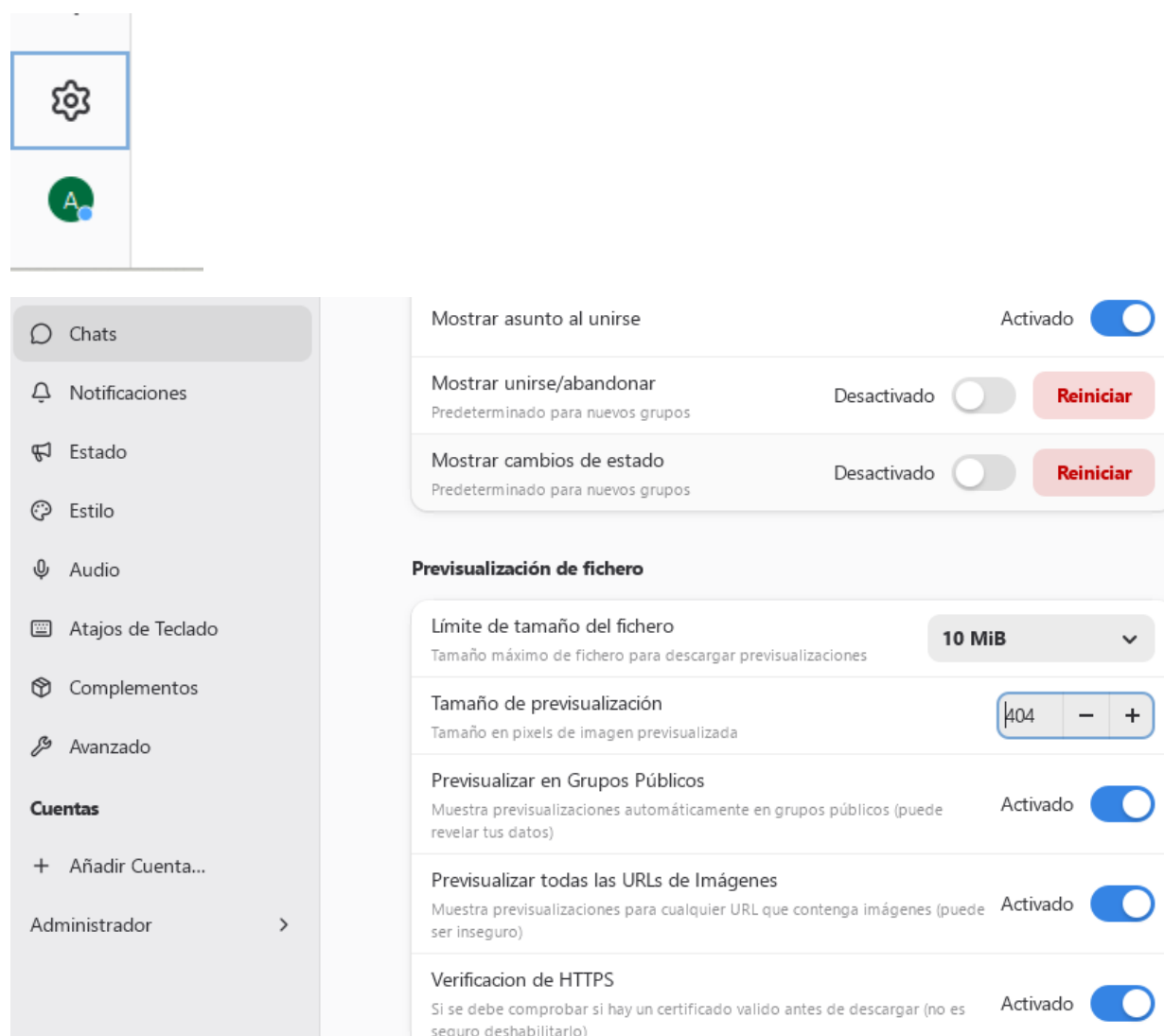
ANEXO:

Gajim en Windows:

Se aconseja usar el cliente Gajim en su versión de Windows debido a los problemas que presenta en Ubuntu.

Una vez instalado, para poder visualizar las imágenes en el chat debemos ir a:

Preferencias (Esquina inferior derecha) → Chats → Previsualización de ficheros: Ahí activaremos ambas opciones de Previsualizar.



Configuración final:

```
admins = { "admin@im07.institutodh.net" }
plugin_paths = {
    "/usr/lib/prosody/modules",
    "/var/lib/prosody/custom_plugins"
}
plugin_server = "https://modules.prosody.im/rocks/"
installer_plugin_path = "/var/lib/prosody/custom_plugins"
modules_enabled = {
    "roster"; -- Allow users to have a roster. Recommended ;)
    "saslauth"; -- Authentication for clients and servers. Recommended if you want to
log in.
    "tls"; -- Add support for secure TLS on c2s/s2s connections
    "dialback"; -- s2s dialback support
    "disco"; -- Service discovery

    "carbons"; -- Keep multiple clients in sync
    "pep"; -- Enables users to publish their avatar, mood, activity, playing music and
more
    "private"; -- Private XML storage (for room bookmarks, etc.)
    "blocklist"; -- Allow users to block communications with other users
    "vcard4"; -- User profiles (stored in PEP)
    "vcard_legacy"; -- Conversion between legacy vCard and PEP Avatar, vcard
    "limits"; -- Enable bandwidth limiting for XMPP connections

    "version"; -- Replies to server version requests
    "uptime"; -- Report how long server has been running
    "time"; -- Let others know the time here on this server
    "ping"; -- Replies to XMPP pings with pongs
    "register"; -- Allow users to register on this server using a client and change
passwords
    -- Admin interfaces
    "admin_adhoc"; -- Allows administration via an XMPP client that supports ad-hoc
commands
    "admin_web"; -- ESTE MODULO ES EL DE ADMINISTRACIÓN POR WEB:
https://modules.prosody.im/rocks/mod_admin_web-86-1.src.rock
    "bosh"; -- https://prosody.im/doc/modules/mod_bosh
    "http"; -- https://prosody.im/doc/http
    "admin_shell"; -- https://prosody.im/doc/modules/mod_admin_shell

    -- Other specific functionality
    "posix"; -- POSIX functionality, sends server to background, enables syslog, etc.
```

```

-- Módulos extras
    "tombstones"; -- Prevent registration of deleted accounts
    "mam"; -- Store recent messages to allow multi-device synchronization
    "checkcerts"; -- https://modules.prosody.im/mod_checkcerts
    "admin_shell"; -- Enable mod_admin_shell
}

modules_disabled = {
}

allow_registration = false -- Cámbialo a 'true' solo cuando necesites crear cuentas. El
módulo "register" debe estar habilitado.

daemonize = true; -- Si lo dejas en "false", cuando cierres la terminal SSH, el servidor de
Prosody se detendrá.
pidfile = "/run/prosody/prosody.pid";
c2s_require_encryption = true -- SEGURIDAD: Obliga a usar cifrado entre cliente y servidor
s2s_require_encryption = true -- SEGURIDAD: Obliga a usar cifrado entre servidores

s2s_secure_auth = true
allow_unencrypted_plain_auth = false

-- Límite velocidad conexión
limits = {
    c2s = {
        rate = "50kb/s";
    };
    s2sin = {
        rate = "100kb/s";
    };
}
authentication = "internal_hashed"
-- archive_expires_after = "2w" -- Remove archived messages after 2 week
log = {
    -- Log files (change 'info' to 'debug' for debug logs):
    debug = "/var/log/prosody/prosody.log";
    error = "/var/log/prosody/prosody.err";
    -- Syslog:
    { levels = { "error" }; to = "syslog"; };
}
certificates = "certs"
VirtualHost "im07.institutodh.net"
    ssl = {
        key = "/etc/prosody/certs/im07.institutodh.net.key";

```

```

    certificate = "/etc/prosody/certs/im07.institutodh.net.crt";
}
Component "conference.im07.institutodh.net" "muc"
modules_enabled = { "muc_mam" }

-- Configuración para subir imagenes???

Component "upload.im07.institutodh.net" "http_file_share"
-- http_file_share_size_limit = 1024 -- 1 KB

--http_file_share_access = {
--"cristobal@im07.institutodh.net", -- this specific user
--}

-- http_file_share_allowed_file_types = { "image/*" }

-- Configuración HTTP y HTTPS

-- http_ports = { 5280 }
-- http_interfaces = { "127.0.0.1", "::1" }

-- https_ports = { 5281 }
-- https_interfaces = { "*", "::" }
--

-- Configuración de tombstones. Se bloquearán al día siguiente de estar borrados.

user_tombstone_expiry = "1d"

-- Configuración mod_mam para sincronización múltiples dispositivos

-- Política de archivado
default_archive_policy = true
-- Guarda TODOS los mensajes por defecto

-- Tiempo de conservación de mensajes
archive_expires_after = "1y"
-- Mantener el historial durante 1 año

-- Intervalo de limpieza (4 horas)
archive_cleanup_interval = 4 * 60 * 60

-- Límite de mensajes por consulta
max_archive_query_results = 50

```



```
-- No archivar estados de escritura, etc.
dont_archive_namespaces = {
    "http://jabber.org/protocol/chatstates";
}

-- Activar archivado solo cuando el cliente lo soporte (opcional)
mam_smart_enable = false

storage = {
    archive = "sql";
}

-- Configuración de checkcert
checkcerts_notify = 7 -- ( in days )

Include "conf.d/*.cfg.lua"
```