Jams User Guide

Savoir-faire Linux
Savoir-faire Linux

Table of content

Getting Started

JAMS is a server application used to enroll Jami clients into an Enterprise context. Currently, JAMS supports 3 sources for user authentication: LDAP, Active Directory and an embedded database.

Obtaining JAMS

The latest version of JAMS can be downloaded at: https://jami.biz/ *(https://jami.biz/)* The source code is available at https://git.jami.net/savoirfairelinux/jami-jams *(https://git.jami.net/savoirfairelinux/jami-jams)*

System Requirements

- Windows, Linux or Mac OS operating system
- Java 11 or higher
- 4 GB RAM
- 1 CPU

JAMS Concepts

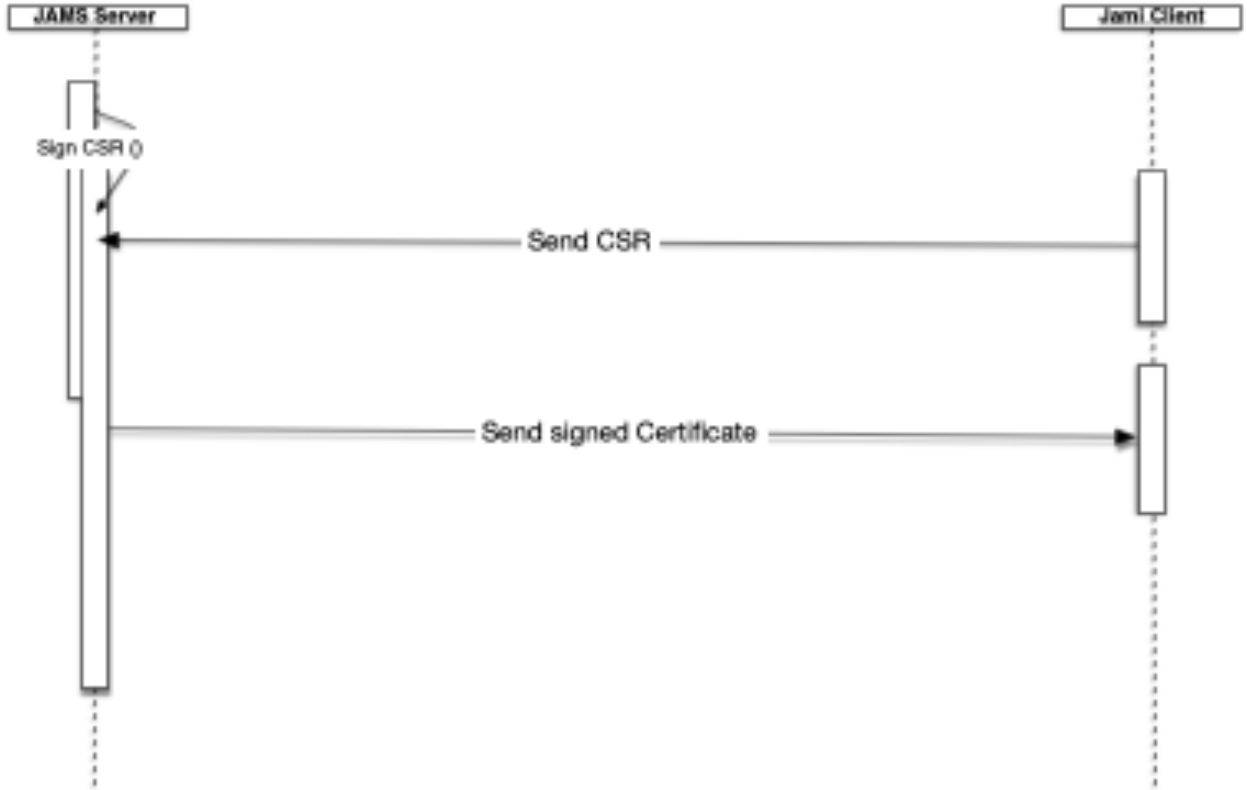
JAMS was built with security in mind, therefore it is intimately related to the X509 certificate management workflows.

The central concepts which are used in JAMS are the Certification Authority (CA) and the Certificate Signing Requests (CSR).

In the JAMS paradigm, a device (Jami client) requests a certificate to the server then presents it to other devices to be recognized as a valid member of the organization. Therefore, JAMS must be provided with a certificate authority in order to work properly.

In order to be completely secure, JAMS does not generate certificates for devices, but instead issues certificates based on a certificate signing request sent to it by the device, therefore removing the need to send a private key over the wire.

The diagram below shows the entire process of how a device enrolls with JAMS:



Getting Started

1. Download the latest version of JAMS from: https://jami.biz/ *(https://jami.biz/)*
2. Unpack the .tar file to a directory of your choice.
3. It is mandatory to run JAMS using a secure SSL connection.

You must have a domain name in order to request a key and a certificate.

Once you have purchased you domain name and pointed it to you server you can proceed to the next step.

You can purchase a pair of key certificate from any online provider such as GoDaddy, OVH, HostGator , etc. We recommend getting a free pair using Let's encrypt.

In order to generate a pair of key certificate you can use Certbot using instructions in the following page https://certbot.eff.org/.

You can choose the web server software and operating system to get specific instructions.

Here is an example for an Nginx web server on Ubuntu 20.04: https://certbot.eff.org/lets-encrypt/ubuntufocal-nginx

Install Certbot using snap: sudo snap install --classic certbot

Ensure that the cerbot command can be run: sudo ln -s /snap/bin/certbot /usr/bin/certbot

In order to get a certificate execute: sudo certbot certonly and follow instructions.

The Certificate and Key are generated in a specific folder, please see the output from Certbot to locate them.

We need to copy them in the current folder where our jams-launcher.jar file is located.

Current limitation: JAMS does not support reading encrypted private keys which require a password unlock.

1. Navigate to the directory where you have extracted the JAMS package and execute the following command:

```
java -jar jams-launcher.jar PORT SSL_CERTIFICATE SSL_CERTIFICATE_KEY
```

Argument	Details
PORT	The TCP port on which you want JAMS to listen for incoming connections
SSL_CERTIFICATE	The location of the PEM-formatted SSL Certificate file
SSL_CERTIFICATE_KEY	The location of the PEM-formatted key file which is used with the SSL Certificate file from above

An example of the command would be:

```
java -jar jams-launcher 443 server.pem server.key
```

Please note that any port above 1024 can be safely used to run JAMS.

Step 1: create your administrator account

This account will have administrative control and the rights to manage your users and group of Jami users.

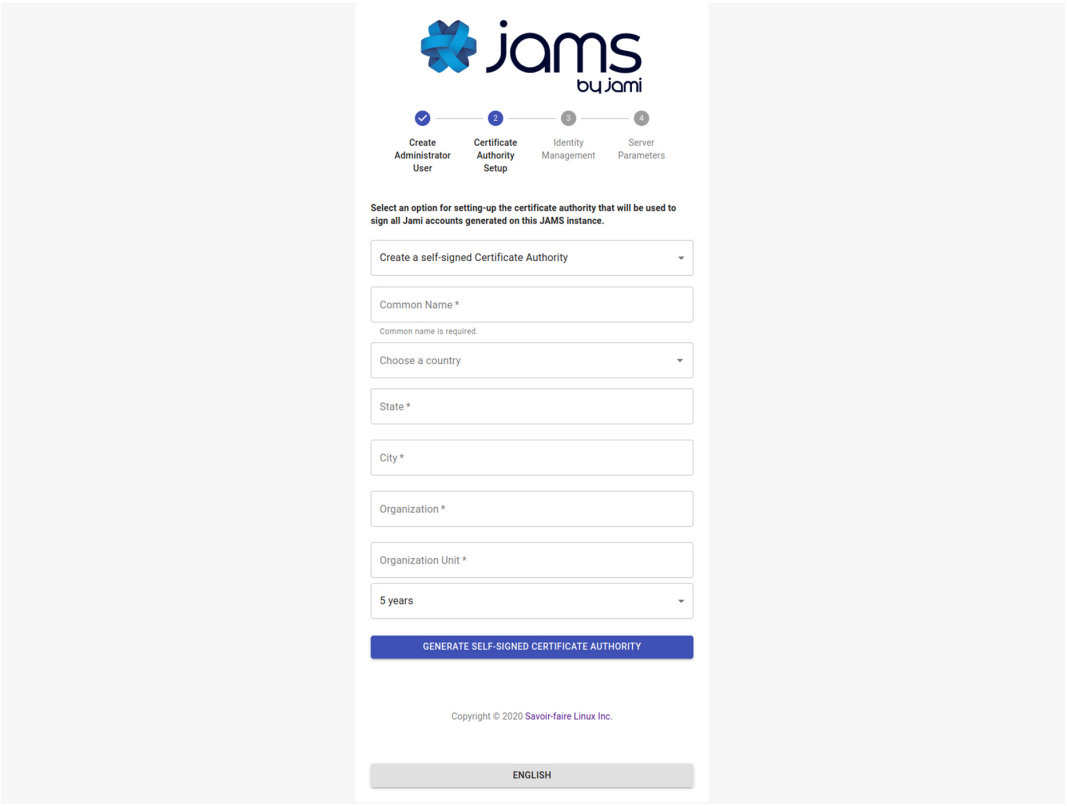
alt text

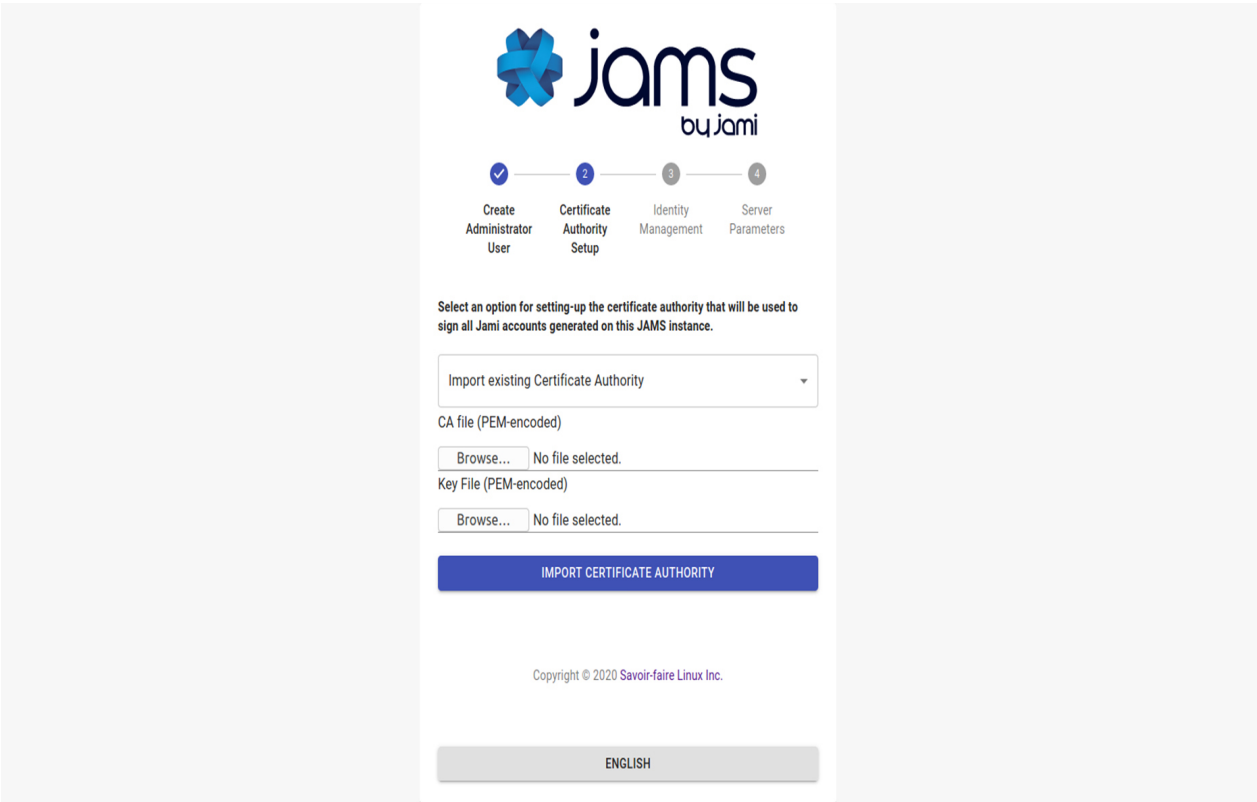
Step 2: setup the Certification Authority

The second step is to define your Certification Authority.

Important: a CA is not a server-side ssl certificate, it is a certificate which has the power to issue other certificates. Do not use the import option unless your company's security officer has issued you a CA certificate. Most commercially available certificates (i.e. those issued by

GoDaddy, Let's Encrypt, etc...) are not CA certificates. If you are an end-user we highly recommend you use to create a self-signed CA option as providing an incorrect certificate type will lead to a non-functional server.





This certificate will be used to sign the enrollment requests which come from Jami devices. If you are not familiar with the X509 standard, we highly recommend you read the following articles to get familiar with the processes and practices which surround it:

https://www.securew2.com/blog/public-key-infrastructure-explained/ *(https://www.securew2.com/blog/public-key-infrastructure-explained/)* https://cheapsslsecurity.com/blog/understanding-the-role-of-certificate-authorities-in-pki/ *(https://cheapsslsecurity.com/blog/understanding-the-role-of-certificate-authorities-in-pki/)*

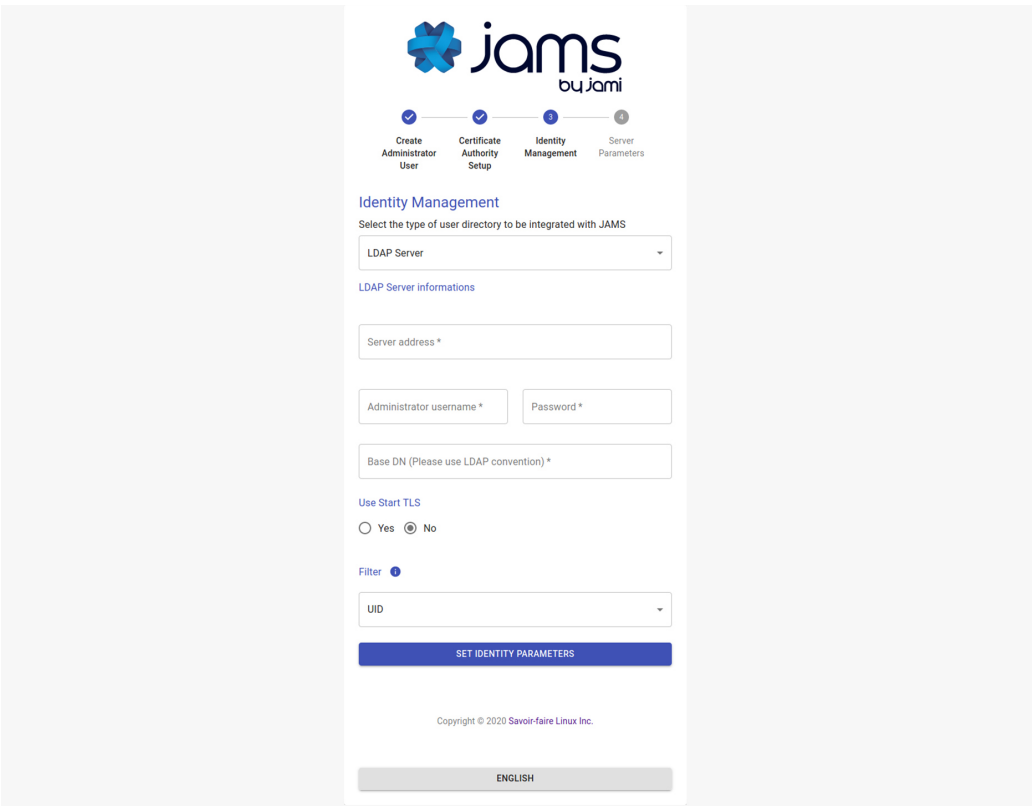
Step 3: setup the user database

JAMS supports 3 different sources for the authentication of users:

- LDAP-compatible directory (such as OpenLDAP)
- Microsoft Active Directory
- Local embedded database

Option 1: LDAP authentication

If your company provides you with LDAP directory for user management, you will need to know its access information and an automated account which has read-only rights to do use look-ups.

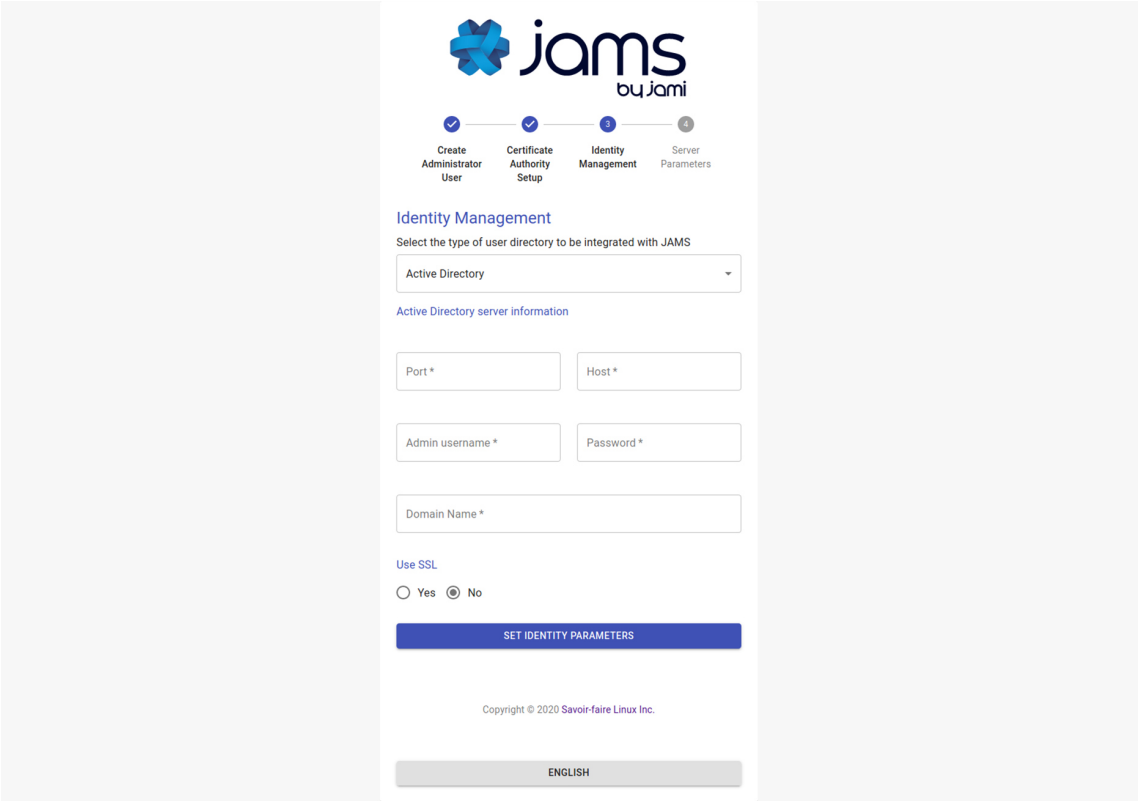


Your admin should provide you most of this information but we do provide a detailed overview over each field in case you need some extra help:

Field	Details
Use StartTLS	Your LDAP server can be configured to use either TLS/STARTTLS or PLAIN sockets, if STARTTLS is used you should mark this as true
Server Address	The address of your server with respect to the JAMS server, your LDAP does not need to be publicly accessible but should be accessible to JAMS. You should have either `ldap://` or `ldaps://` preceding the address.
Port	The port on which the LDAP server is listening for requests (usually 389 for PLAIN/STARTTLS and 636 for SSL/TLS)
Administrator Username	This is **NOT** the LDAP's administration account credentials, but the credentials of the account which has Read permissions to the LDAP database in order to lookup users. The format is generally `cn=bot,ou=robots,dc=domain,dc=org`
Password	The password used by the account above.
BaseDN	The base realm where the users accounts are located, in most cases it is `ou=users,dc=company,dc=org`

Option 2: Microsoft Active Directory

If your company provides you with Active Directory for user management, you will need to know its access information and an automated account which has read-only rights to do use look-ups.

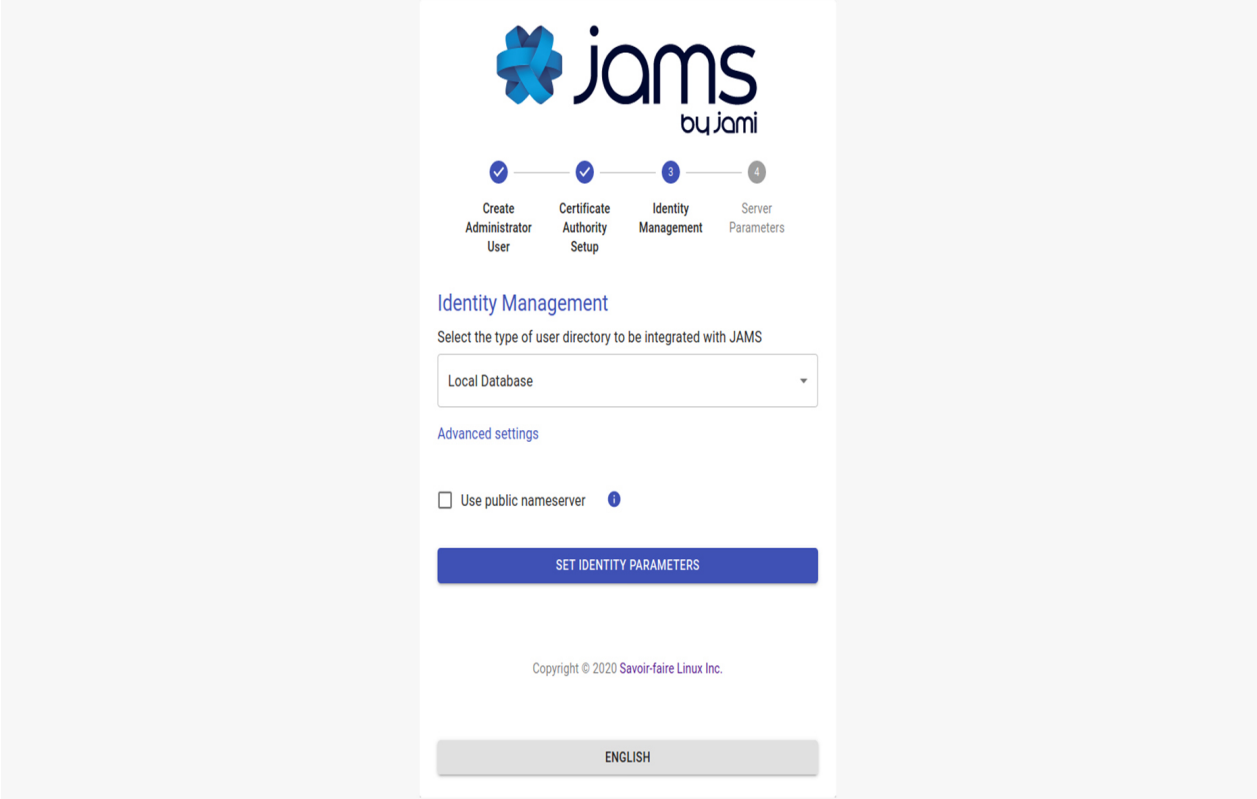


Your admin should provide you most of this information but we do provide a detailed overview over each field in case you need some extra help:

Field	Details
Port	The port on which Active Directory is listening (generally it is either 389 or 636)
Host	The address of your server with respect to the JAMS server, your Active Directory does not need to be publicly accessible but should be accessible to JAMS.
Administrator Username	This is **NOT** the Active Directory's administration account credentials, but the credentials of the account which has Read permissions to the Active Directory database in order to lookup users. The format is generally `cn=bot,ou=robots,dc=domain,dc=net`
Password	The password used by the account above.
Use SSL	Whenever this server uses SSL for data transmission
Domain Name	This is the legacy-formatted Windows Domain Name (i.e. `WINDOMAIN`)

Option 3: local embedded database

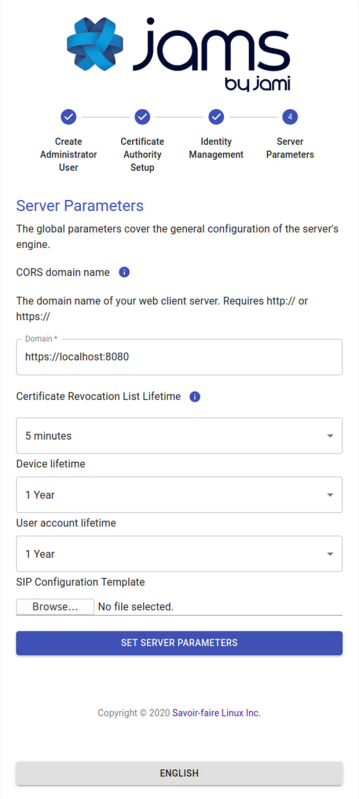
The local database does not require any additional configuration, everything in the process is automated. This option allows you to create Jami users on the fly directly from the JAMS interface.



Advanced settings: by default, the option "Use public nameserver" is disabled. Usernames of your Jami users will not be stored on the public Jami nameserver and your users will only be able to communicate with users from your organization.

If you want your users to be searchable by external users and allow them to communicate with any Jami users, and not only the one from your organization, enable this option.

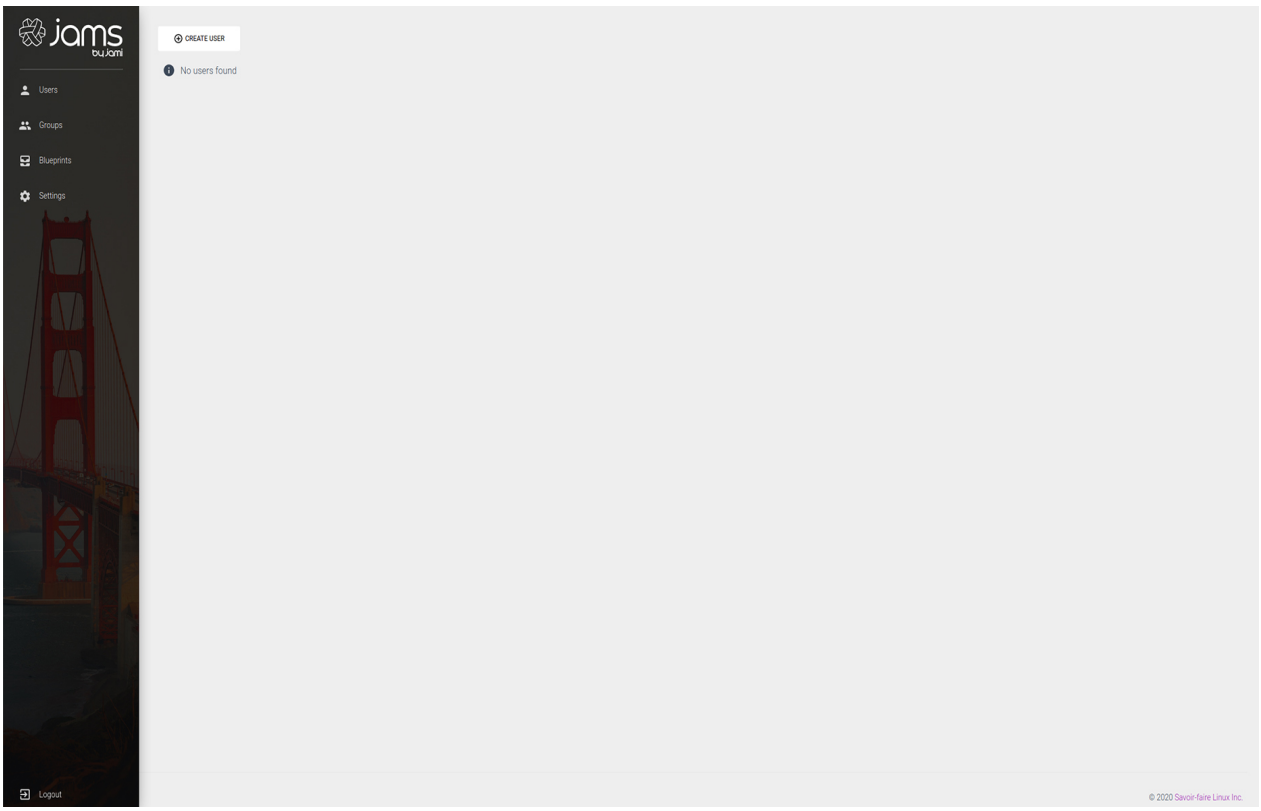
Step 4: setup the server parameters



Parameter	Details
CORS Domain Name	The domain on which the JAMS client and administration UI will be running.
Certificate Revocation List Lifetime	The frequency at which the CRL is updated in memory
Device Lifetime	How long a device's certificate is valid before being considered stale and requiring re-enrollment
User Account Lifetime	How long a user account is valid before being considered stale and requiring re-enrollment

Important The *CORS Domain Name* corresponds to the web address used to access the Web UI. By default, it is set to the same URL as the one where you deploy JAMS. Only set a different URL if the Web UI has a different URL to the one where JAMS is deployed.

Click on "Set Server Parameters" to finalize the configuration. You will be redirected to the JAMS interface.



If you have configured JAMS with your LDAP or Active Directory, the list of users should of your organization should be visible in JAMS. If you have selected the local embedded database, you can now start creating new users by clicking on "Create User".

Admin Guide

By default JAMS runs an embedded tomcat server visible on port 8080, however this is not practical for many reasons. This guide is designed to help you setup Jams to run in a production environment.

JAMS & Nginx

It is generally not recommended to expose JAMS directly to the outside world and while it is required to run JAMS in SSL mode, we usually recommend users to place it behind Nginx or a similar web server which proxies requests between the outside world and Jams.

The following is an example map of how you could configure JAMS behind Nginx (the process would be similar if you wanted to use any other type of proxy solution):

alt text

The IP 10.10.0.1 is random, and should be seen as an example.

Typically you would add a new site called `jams-site.conf` to your nginx configurations which would contain the following entries if you wanted to place an SSL certificate at the Nginx level:

```
server {
        listen 443 ssl;
        listen [::]:443 ssl;
        ssl on;
        ssl_certificate /etc/certificates/mycertificate.pem
        ssl_certificate_key /etc/certificates/mycertificatekey.pem
        client_max_body_size 100M;
        server_name jams.mycompany.com;
        location / {
                proxy_pass              http://10.10.0.1:8080/;
                proxy_set_header        X-Real-IP  $remote_addr;
                proxy_set_header        Host $http_host;
        }
}
```

This is the preferred setup method by most admins, as local traffic is usually run unencrypted since it is usually either inter-VM connection, a VLAN or another dedicated link.

Troubleshooting and resetting

If you ever need to restart from 0 (i.e. reset everything and drop existing data) you can do so by deleting the following files in the distribution folder (/jams):

The internal jams folder: /jams/jams
derby.log
oauth.key
oauth.pub
config.json

This will reset the server to its original state and you will be able to run the configuration wizard again. Before performing this operation, please make sure to shutdown the server.

Running JAMS as Windows Service

Download and install JAMS

Visit https://jami.biz/ *(https://jami.biz/)* and download JAMS.

Extract JAMS to c:\jams

Download and install JDK 11

Download JDK 11 from https://www.oracle.com/java/technologies/javase-jdk11-downloads.html (choose the corresponding architecture of your VM)

Install it using the install wizard.

Download openssl to generate a key and a certificate

Download OpenSSL from https://kb.firedaemon.com/support/solutions/articles/4000121705 (or choose another source https://wiki.openssl.org/index.php/Binaries)

Once downloaded extract it to c:\openssl then create a folder bin inside c:\openssl\bin

Create a new file inside bin named openssl.cnf (make sure that the file extension is .cnd and not .cnd.txt) and copy past the following default configuration http://www.flatmtn.com/article/setting-openssl-create-certificates.html

```
#
# OpenSSL configuration file.
#
```

```
# Establish working directory.

dir                 = .

[ ca ]
default_ca              = CA_default

[ CA_default ]
serial                  = $dir/serial
database                = $dir/certindex.txt
new_certs_dir               = $dir/certs
certificate             = $dir/cacert.pem
private_key             = $dir/private/cakey.pem
default_days                = 365
default_md              = md5
preserve                = no
email_in_dn             = no
nameopt                 = default_ca
certopt                 = default_ca
policy                  = policy_match

[ policy_match ]
countryName             = match
stateOrProvinceName         = match
organizationName            = match
organizationalUnitName          = optional
commonName              = supplied
emailAddress                = optional

[ req ]
default_bits                = 1024          # Size of keys
default_keyfile             = key.pem       # name of generated keys
default_md              = md5               # message digest algorithm
string_mask             = nombstr       # permitted characters
distinguished_name          = req_distinguished_name
req_extensions              = v3_req

[ req_distinguished_name ]
# Variable name             Prompt string
#-------------------------   ----------------------------------
0.organizationName          = Organization Name (company)
organizationalUnitName          = Organizational Unit Name (department, divisio
emailAddress                = Email Address
emailAddress_max            = 40
```

```
localityName                = Locality Name (city, district)
stateOrProvinceName         = State or Province Name (full name)
countryName             = Country Name (2 letter code)
countryName_min             = 2
countryName_max             = 2
commonName              = Common Name (hostname, IP, or your name)
commonName_max              = 64

# Default values for the above, for consistency and less typing.
# Variable name             Value
------------------------      ------------------------------
0.organizationName_default      = My Company
localityName_default            = My Town
stateOrProvinceName_default     = State or Providence
countryName_default         = US

[ v3_ca ]
basicConstraints            = CA:TRUE
subjectKeyIdentifier            = hash
authorityKeyIdentifier          = keyid:always,issuer:always

[ v3_req ]
basicConstraints            = CA:FALSE
subjectKeyIdentifier            = hash
```

Add OpenSSL to System Environment variables

Go to Edit the system environment variables -> Environment Variables, then in System variables edit Path and add c:\openssl\

Configure OpenSSL

Execute the following command to set the path to OpenSSL configuration.

set OPENSSL_CONF=c:\openssl\bin\openssl.cnf

Open the command prompt and cd c:\jams ans generate the Key and Certificate:

openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout server.key -out server.pem

Follow the wizard.

Once the key and certificate are generated execute the dir command you should see an output like this:

```
c:\jams>dir
 Volume in drive C has no label.
 Volume Serial Number is BC94-9EF2

 Directory of c:\jams

2020-11-10  12:38 PM

               .
    2020-11-10  12:38 PM
                   ..
        2020-10-22  10:56 AM         5,186,016 jams-launcher.jar
        2020-10-22  10:56 AM        33,413,882 jams-server.jar
        2020-11-10  11:53 AM
                      libs
            2020-11-10  12:34 PM             1,732 server.key
            2020-11-10  12:38 PM             1,336 server.pem
            2020-10-22  04:05 PM         2,047,932 userguide.pdf
                           5 File(s)     40,650,898 bytes
                           3 Dir(s)  93,365,936,128 bytes free
```

Now execute the following command tot start JAMS

java -jar jams-launcher.jar PORT_NUMBER (eg. 8443 or 443) server.pem server.key

Open a navigator on the server and visite https://localhost:443 or https://localhost:8443 to validate that it's working.

Type CTRL + C to close the application

Expose your localhost to the internet

Click on Windows ans search for Windows Defender Firewall with Advanced Security.

Right click on Inbound Rules and click on New Rule...

Select Port click next and specify the port you want to use example 443 or 8443.

Click next and select Allow the connection and click next.

Leave all of Domain Private and Public select and click next.

Name you Rule JAMS Inbound and click Finish

Now right click on Outbound Rules and click on New Rule...

Select Port click next and specify the port you want to use example 443 or 8443.

Click next and select Allow the connection and click next.

Leave all of Domain Private and Public select and click next.

Name you Rule JAMS Outbound and click Finish.

You are all set. You can now visit your application through the server domain name or IP address on port 443 or 8443.

Create a JAMS Windows Service (Embed Tomcat Server Windows Service) to start JAMS with the server

In order to create a JAMS Windows Service you can use the tool NSSM provided on http://nssm.cc/download https://github.com/kirillkovalenko/nssm *(https://github.com/kirillkovalenko/nssm)*

Once downloaded open a command prompt and change directory to nssm-2.24\win64 then execute:

```
nssm.exe install JAMS
```

A GUI interface will open.

In the Path field specify the path to the Java executable example:

```
"C:\Program Files\Common Files\Oracle\Java\javapath\java.exe".
```

In the Startup directory put the

```
"C:\jams" installation folder path.
```

In the last field Arguments add the following arguments:

```
-classpath "c:\jams" -jar jams-launcher.jar PORT_NUMBER server.pem server.key
```

where PORT_NUMBER is the port number you want to use to serve the application example 443 or 8443

Now your JAMS application will start with the server.

Source: https://medium.com/@lk.snatch/jar-file-as-windows-service-bonus-jar-to-exe-1b7b179053e4 *(https://medium.com/@lk.snatch/jar-file-as-windows-service-bonus-jar-to-exe-1b7b179053e4)*

Running JAMS as a Linux Service

Running JAMS as a Linux Service is fairly straightforward with systemd - you simply created a service unit file with the following structure:
[Unit] Description=JAMS Server

[Service] Type=simple WorkingDirectory=[DIRECTORY WHERE JAMS WAS UNZIPPED] ExecStart=/usr/bin/java -jar [DIRECTORY WHERE JAMS WAS UNZIPPED]/jams-launcher.jar PORT SSL_CERTIFICATE SSL_CERTIFICATE_KEY

[Install] WantedBy=multi-user.target The parameters PORT, SSL_CERTIFICATE and SSL_CERTIFICATE_KEY are optional (however, PORT can be used alone whereas the SSL_CERTIFICATE comes in pair with SSL_CERTIFICATE_KEY)

Client Guide

Depending on your operating system, we have included the tutorial on how to connect to the management server from the Windows, MacOS, Android and iOS clients.

For the purposes of this tutorial, we assume that

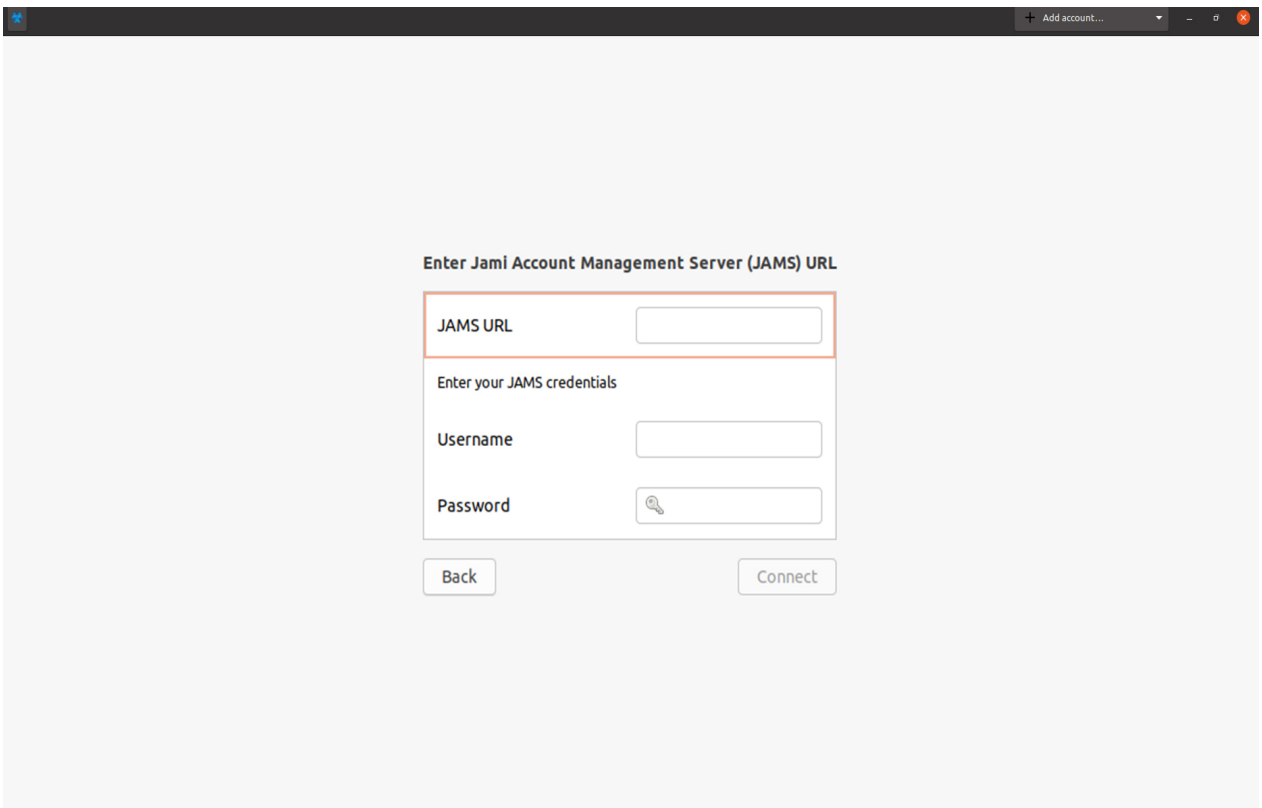
1. The server and the device trying to connect are either
    1. On the same network
    2. The server is publicly accessible to the outside world
2. You have a valid username/password pair to connect to the server

Connect from a Linux device

Open Jami, go to the login page. Click on "Advanced":

alt text

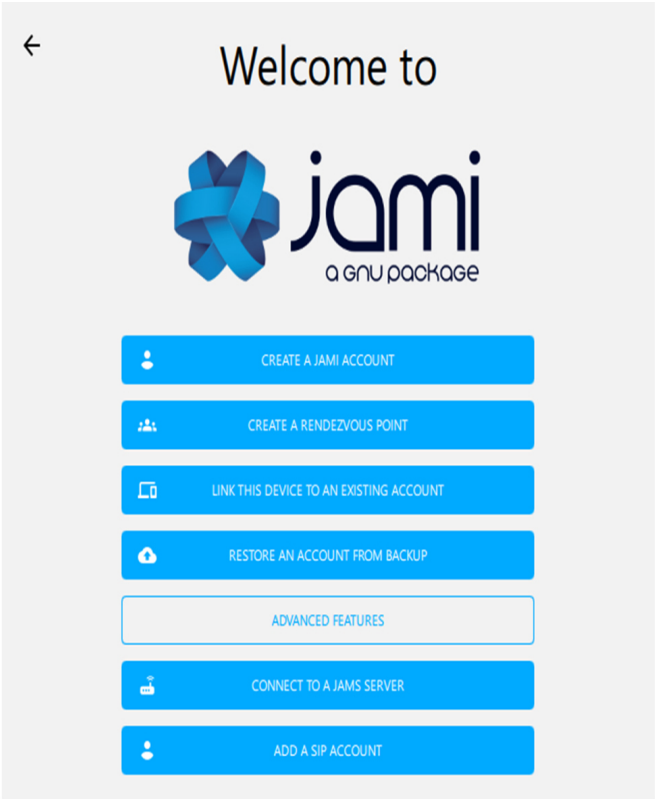
Select the option **"Connect to a JAMS server"** which will lead you to the following screen:



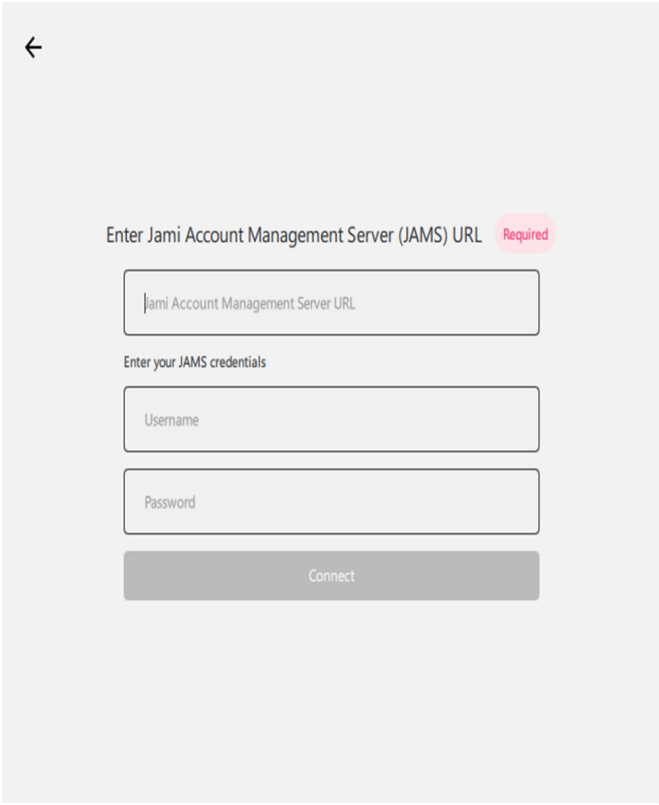
The **Jami Account Management Server URL** in this case would be the DNS address of your server and the username and password which correspond to your account. If you have configured the server with an LDAP/AD backend, it would be your LDAP/AD username and password.

Connect from a Windows device

Open Jami, go to the login page. Click on "Advanced":



Select the option **"Connect to a JAMS server"** which will lead you to the following screen:



The **Jami Account Management Server URL** in this case would be the DNS address of your server and the username and password which correspond to your account. If you have configured the server with an LDAP/AD backend, it would be your LDAP/AD username and password.

Connect from a macOS device

Open Jami, go to the login page. Click on "Advanced":

alt text

Select the option **"Connect to account manager"** which will lead you to the following screen:

alt text

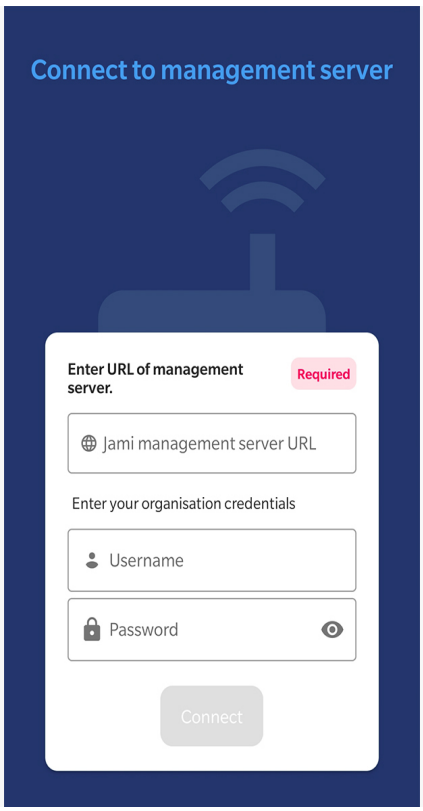
The **Jami Account Management Server URL** in this case would be the DNS address of your server and the username and password which correspond to your account. If you have configured the server with an LDAP/AD backend, it would be your LDAP/AD username and password.

Connect from an Android device

Open Jami, go to the login page.

alt text

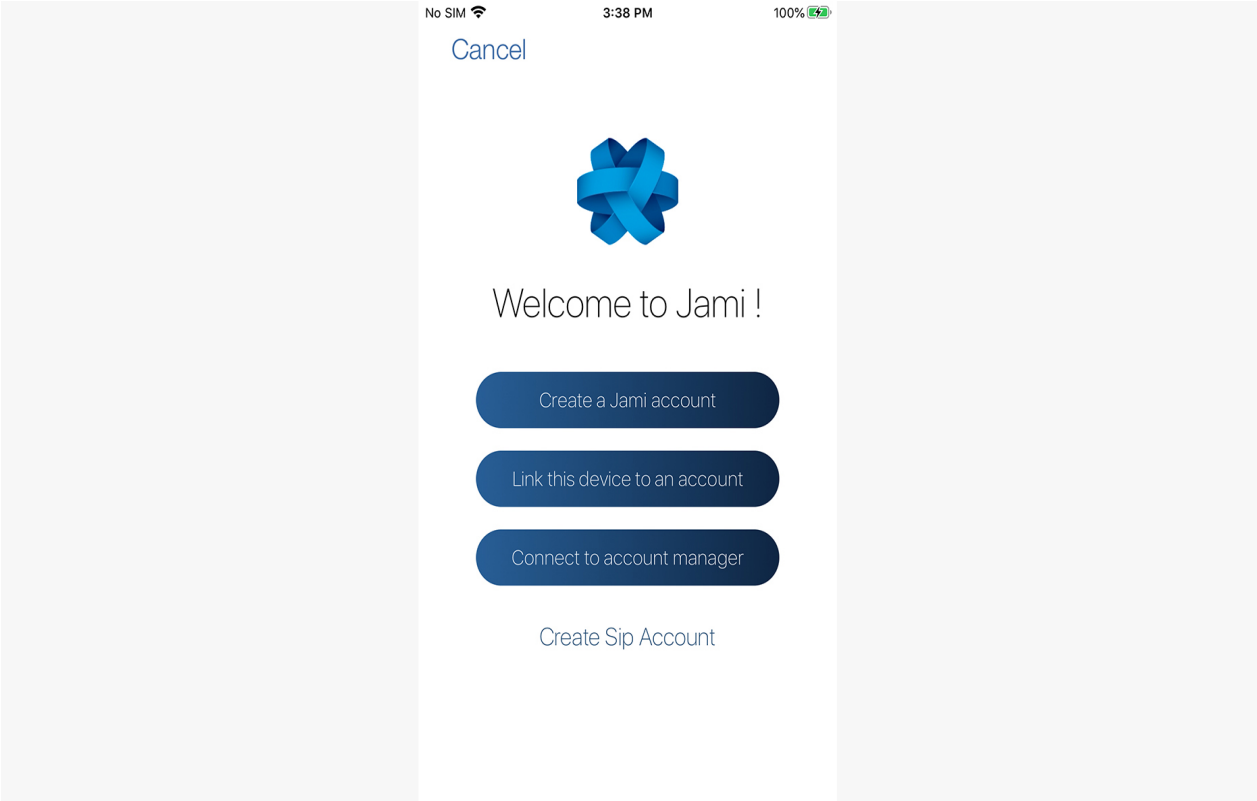
Select the option **"Connect to management server"** which will lead you to the following screen:



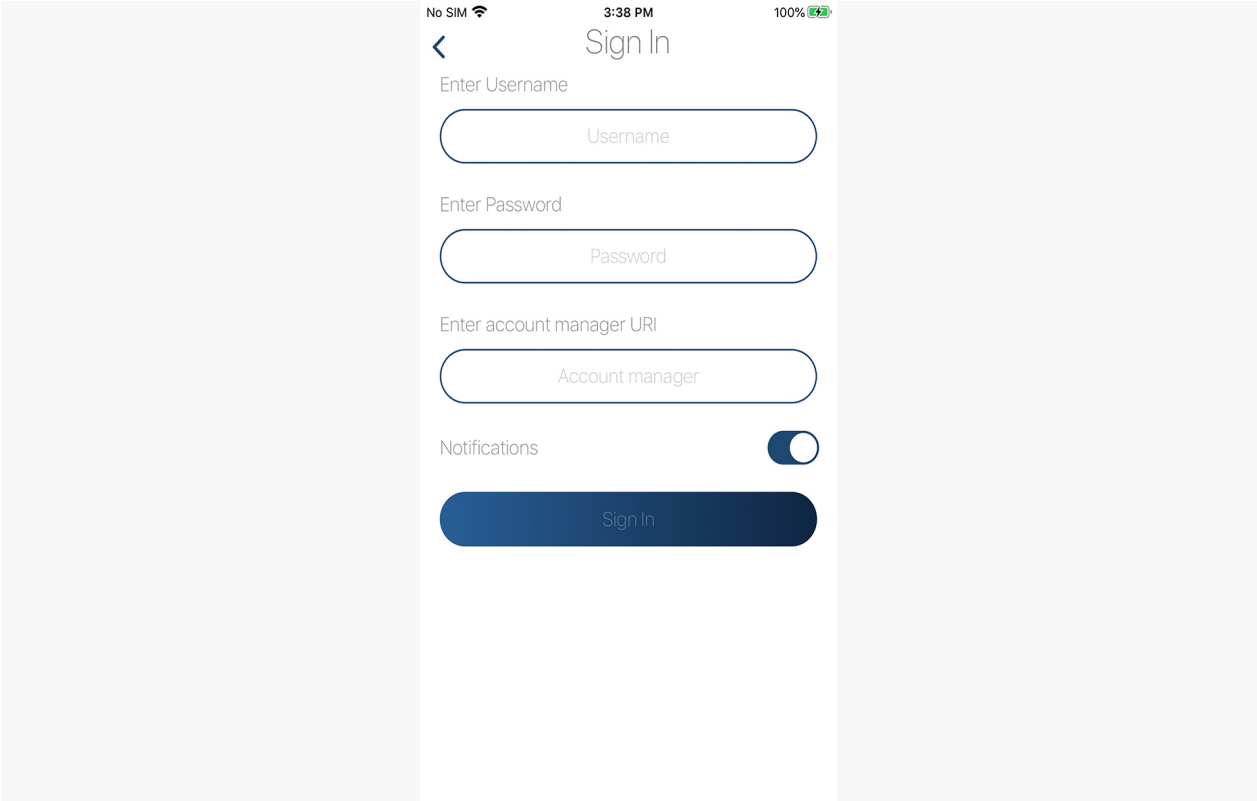
The server in this case would be the DNS address of your server and the username and password which correspond to your account. If you have configured the server with an LDAP/AD backend, it would be your LDAP/AD username and password.

Connect from an iOS device

Open Jami, go to the login page.



Select the option **"Connect to account manager"** which will lead you to the following screen:



The server in this case would be the DNS address of your server and the username and password which correspond to your account. If you have configured the server with an LDAP/AD backend, it would be your LDAP/AD username and password.