

Jams User Guide

CC-BY-SA

Table of Contents

Getting Started

- [Getting Started](#)
 - [Obtaining Jams](#)
 - [System Requirements](#)
 - [Jams Concepts](#)
 - [Getting Started](#)
 - [Step 1: Create your admin account](#)
 - [Step 2: Setup the Certification Authority](#)
 - [Step 3: Setup the user database](#)
 - [LDAP Authentication source](#)
 - [Microsoft Active Directory](#)
 - [Local Embedded Database](#)
 - [Step 4: Server Parameters](#)
 - [Troubleshooting and resetting](#)

Admin Guide

- [Admin Guide](#)
 - [Jams & Nginx](#)
 - [Running Jams with SSL](#)
 - [Running Jams as a Linux Service](#)
 - [Running Jams as a Windows Service](#)

Client Guide

- [Client Guide](#)
 - [Connecting using Android](#)
 - [Connecting using Mac OS](#)
 - [Connecting using Windows](#)

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 current alpha build of JAMS can be downloaded at: <https://>

System Requirements

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

Jams Concepts

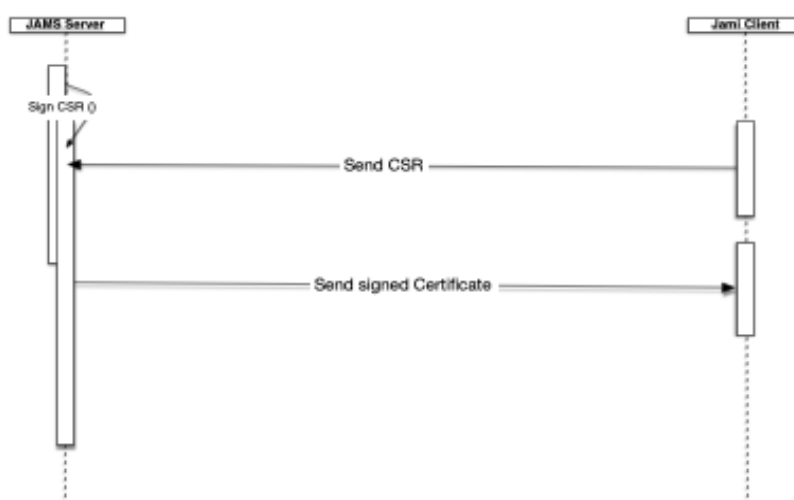
Jams was built with security in mind, therefore it is intimately related to the X509 certificate management workflows.

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) basically requests the server to issue a certificate to it in order to present it to other devices which lets them recognize the device as a valid member of the organization, therefore Jams MUST be provided with a certificate authority in order to function correctly. Please note that a CA is NOT a standard SSL server certificate, as they do not have the permission to issue certificates.

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

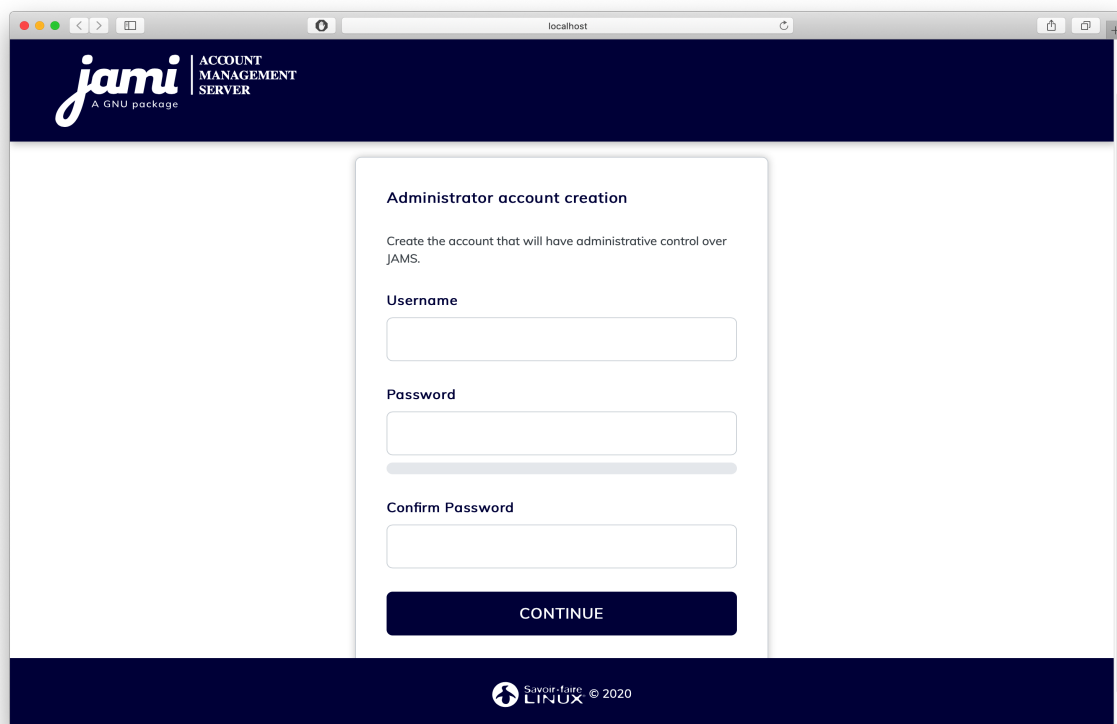
Download the latest version from: <https://dl.jami.net/jams/jams-alpha.zip>

Unpack the ZIP file to a directory of your choice.

To run the server, navigate to the directory where you have extracted the Jams package and execute `java -jar jams-launcher.jar`

Step 1: Create your admin account

This account will be used for administrative purposes, it is used to browse the user database, removing devices and performing other basic administrative tasks.



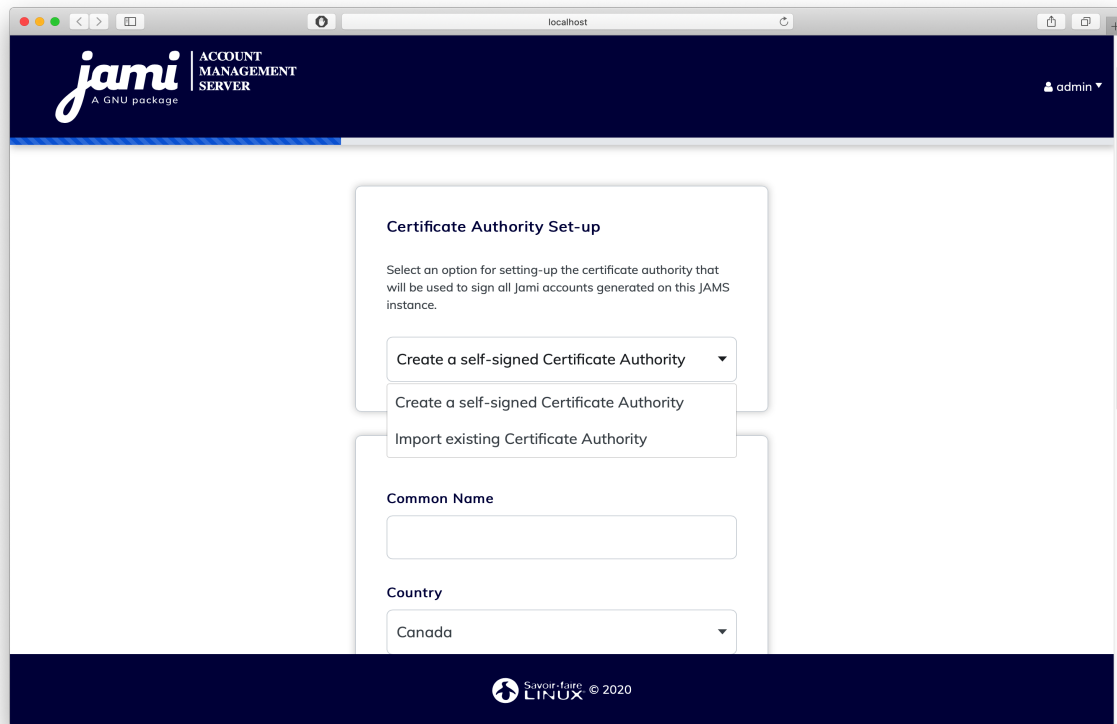
The screenshot shows a web browser window displaying the Jami Account Management Server interface. The header features the Jami logo and the text "ACCOUNT MANAGEMENT SERVER" and "A GNU package". The main content area is titled "Administrator account creation" and includes the instruction "Create the account that will have administrative control over JAMS." Below this, there are three input fields: "Username", "Password", and "Confirm Password". A "CONTINUE" button is located at the bottom of the form. The footer of the page shows the "Swing Linux" logo and the copyright notice "© 2020".

Step 2: Setup the Certification Authority

The second step is to define your Certification Authority.

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 COMMERCIALY AVAILABLE CERTIFICATES (I.E. THOSE ISSUED BY GODADDY, LETSENCRYPT, ETC...) ARE NOT CA CERTIFICATES. IF YOU ARE AN END-USER WE HIGHLY RECOMMEND YOU USE THE CREATE A SELF-SIGNED CA OPTION

AS PROVIDING AN INCORRECT CERTIFICATE TYPE WILL LEAD TO A NON-FUNCTIONAL SERVER!!!



The screenshot shows a web browser window with the URL 'localhost'. The page header features the 'jami' logo (A GNU package) and 'ACCOUNT MANAGEMENT SERVER' on the left, and a user profile 'admin' on the right. The main content area is titled 'Certificate Authority Set-up' and contains the following elements:

- A sub-header: 'Certificate Authority Set-up'
- Instructional text: 'Select an option for setting-up the certificate authority that will be used to sign all Jami accounts generated on this JAMS instance.'
- A dropdown menu with the selected option: 'Create a self-signed Certificate Authority'.
- Below the dropdown, the text 'Create a self-signed Certificate Authority' is repeated.
- A link: 'Import existing Certificate Authority'.
- A 'Common Name' text input field.
- A 'Country' dropdown menu with 'Canada' selected.

The footer of the page displays the 'Savoir-faire LINUX' logo and '© 2020'.

This certificate will be used to sign the enrollement 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://cheapsslsecurity.com/blog/understanding-the-role-of-certificate-authorities-in-pki/>

Step 3: Setup the user database

Currently, Jams supports 3 sources of authentication of users:

1) LDAP-compatible directory (such as OpenLDAP) 2) Microsoft Active Directory 3) Local embedded database

The screenshot shows the Jami Account Management Server web interface. The header includes the Jami logo and 'ACCOUNT MANAGEMENT SERVER A GNU package'. The user is logged in as 'admin'. The main content area displays the 'Users Directory Selection' form, which prompts the user to 'Select the type of user directory to be integrated with JAMS.' The form includes a dropdown menu currently set to 'LDAP Server', with a list of options: 'LDAP Server', 'Active Directory', and 'Local HSQL Database'. Below the dropdown are radio buttons for 'Yes' and 'No'. Further down are input fields for 'Server Address', 'Administrator Username', and 'Password'. The footer features the 'Savoir-faire LINUX' logo and the year '© 2020'.

LDAP Authentication source

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

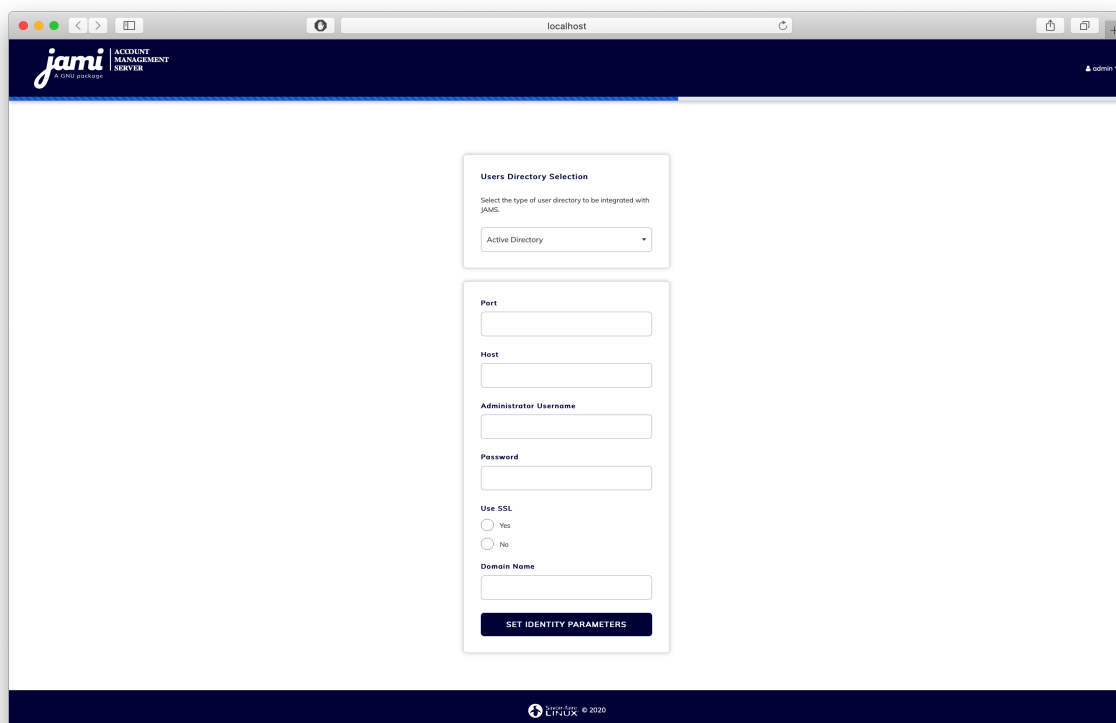
This screenshot shows the same Jami Account Management Server web interface, but with more fields visible in the 'Users Directory Selection' form. The dropdown menu is still set to 'LDAP Server'. Below the 'Yes/No' radio buttons, there are input fields for 'Server Address', 'Administrator Username', and 'Password'. Below these is a field for 'Base DN (Please use LDAP convention)'. At the bottom of the form, there is a 'Filter (This is the field in your LDAP structure which contains the username)' dropdown menu set to 'UID'. A 'SET IDENTITY PARAMETERS' button is located at the bottom of the form. The footer remains the same with the 'Savoir-faire LINUX' logo and '© 2020'.

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 <code>ldap://</code> or <code>ldaps://</code> 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 <code>cn=bot,ou=robots,dc=domain,dc=org</code> |
| Password | The password used by the account above. |
| BaseDN | The base realm where the users accounts are located, in most cases it is <code>ou=users,dc=company,dc=org</code> |

Microsoft Active Directory

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



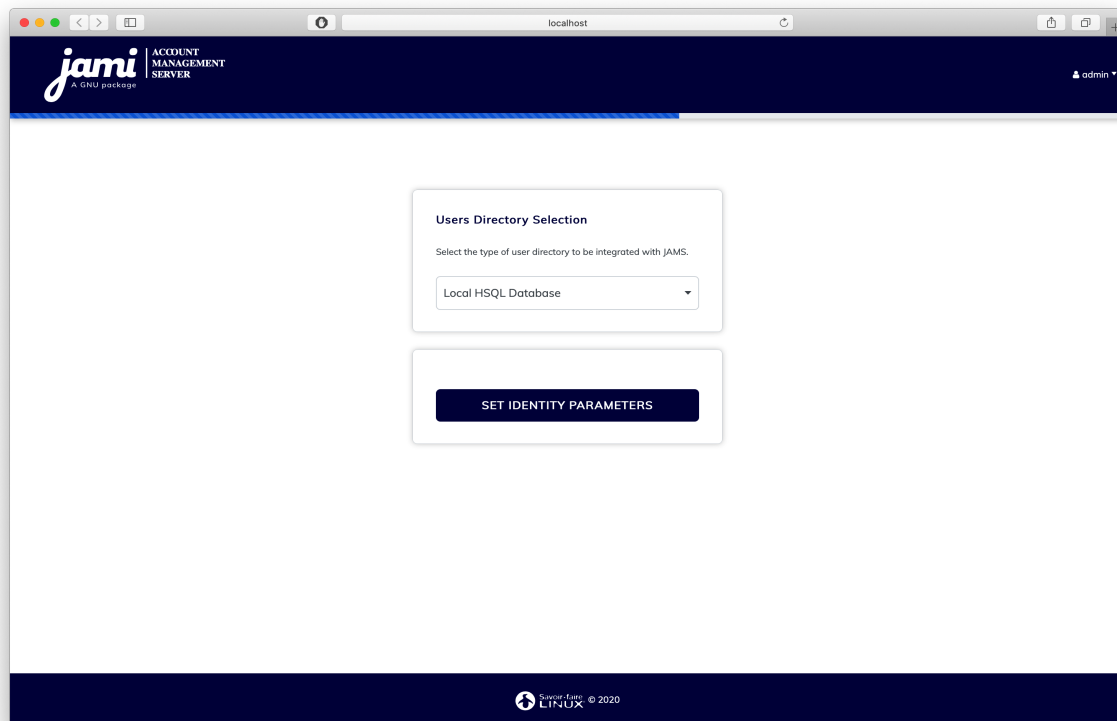
The screenshot shows a web browser window with the address bar set to 'localhost'. The page title is 'jami ACCOUNT MANAGEMENT SERVER'. The main content area contains a form titled 'Users Directory Selection'. The form has a dropdown menu for 'Select the type of user directory to be integrated with JAMS.' with 'Active Directory' selected. Below this are input fields for 'Port', 'Host', 'Administrator Username', and 'Password'. There are radio buttons for 'Use SSL' (Yes/No) and a 'Domain Name' input field. A 'SET IDENTITY PARAMETERS' button is at the bottom of the form. The footer of the page shows the 'JAMIN' logo and '© 2020'.

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 <code>cn=bot,ou=robots,dc=domain,dc=net</code> |
| 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) |

Local Embedded Database

The local database does not require any additional configuration, everything in the process is automated.



Step 4: Server Parameters

The screenshot shows the 'Server Parameters' configuration page in the Jami Account Management Server web interface. The page is titled 'Server Parameters' and includes a description: 'The global parameters cover the general configuration of the server's engine.' The form contains four fields: 'CORS Domain Name' (a text input field), 'Certificate Revocation List Lifetime' (a dropdown menu set to '5 minutes'), 'Device Lifetime' (a dropdown menu set to '1 Month'), and 'User Account Lifetime' (a dropdown menu set to '1 Year'). A 'SET SERVER PARAMETERS' button is located at the bottom of the form. The interface has a dark blue header with the 'jami' logo and 'ACCOUNT MANAGEMENT SERVER' text, and a footer with 'Server Suite © 2020' and 'LINUX'.

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-enrollement

User Account Lifetime

How long a user account is valid before being considered stale and requiring re-enrollement

IMPORTANT NOTICE REGARDING THE FIELD **CORS Domain Name**

Many users have trouble with this part of the installation, to make it explicitly clear, this is web address used to access the Web UI. For example, if you expect users to access the Web UI by visiting the URL `http://jams.mycompany.com` then you should set the **CORS Domain Name** field to `http://jams.mycompany.com`.

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 Jams directory:

```
jams.tmp/  
jams.script
```



```
jams.properties  
tomcat.8080/  
config.json  
keystore.jks  
jams.log  
tmpjar/  
jams.lock
```

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

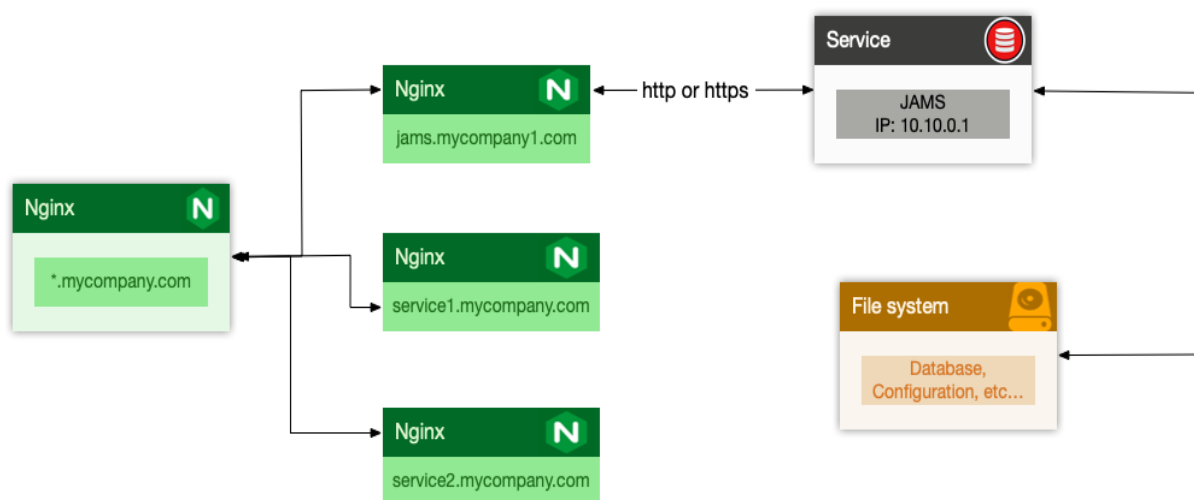
Admin Guide

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 possible 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 proxying solution):



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 ran unencrypted since it is usually either inter-VM connection, a VLAN or another dedicated link.

Running Jams with SSL

If necessary it is possible to run Jams with SSL. In this case, you need to overload the command-line arguments when starting the server.

```
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.jar 8443 /opt/mycert.pem /opt/mycertkey.pem`

Current Limitation Warning: Jams does not support reading encrypted private keys which require a password unlock.

There are only two possible cases when you would want to run Jams with SSL - we do not recommend this method:

1. Your local traffic is exposed to many actors (employees, contractors) and there is no possibility to isolate it
2. You want to expose Jams directly to the external world because proxying is not an option

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-l

[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)

Running Jams as a Windows Service

In progress.

Client Guide

Depending on your operating system, we have included the tutorial on how to connect to the management server using the Windows, Android and Mac OS X 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


Connecting using Android


Upon opening Jami, you will be offered the following screen




A Jami account allows you to reach people securely in peer to peer through a fully distributed network.

 CREATE A JAMI ACCOUNT >

 CONNECT FROM OTHER DEVICE >

 CONNECT FROM BACKUP >

 CONNECT TO MANAGEMENT SERVER >

You should select the option "**CONNECT TO MANAGEMENT SERVER**" which will lead you to the following screen:

Connect to management server

Enter URL of management server

Jami management server URL

Enter your organisation credentials

Username

Password

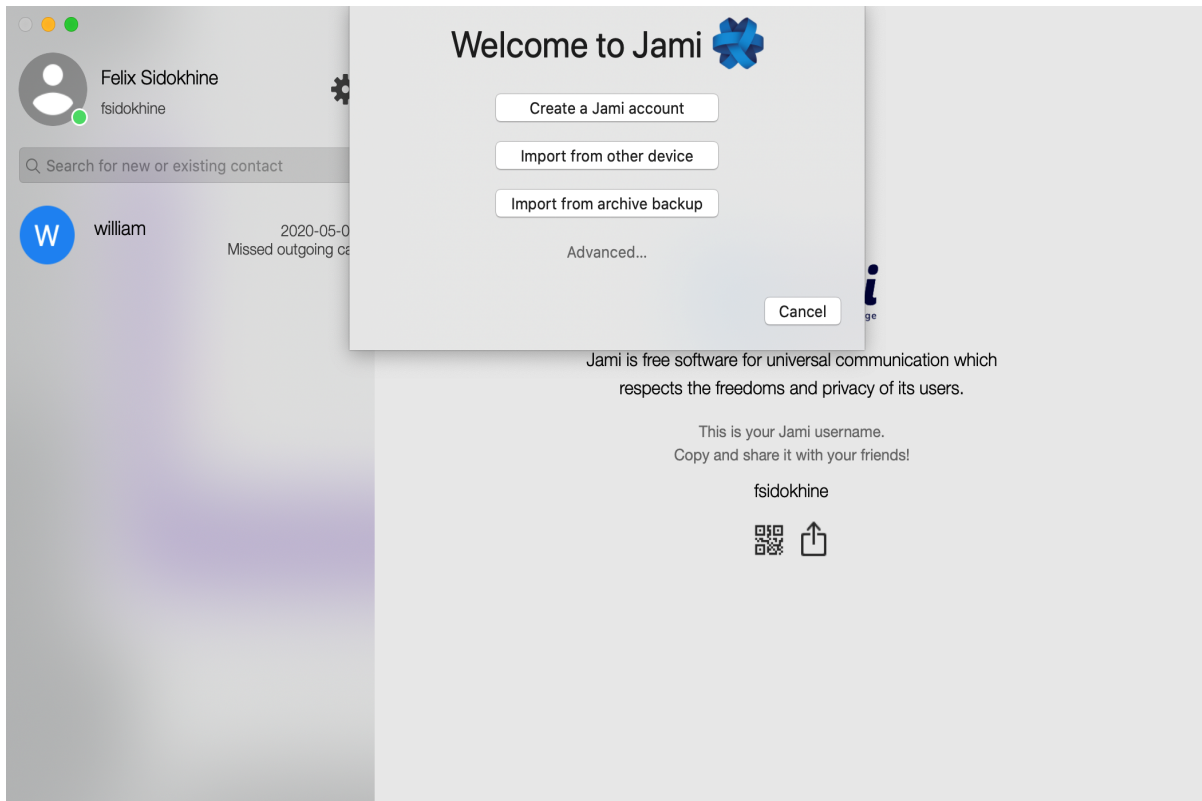


CONNECT

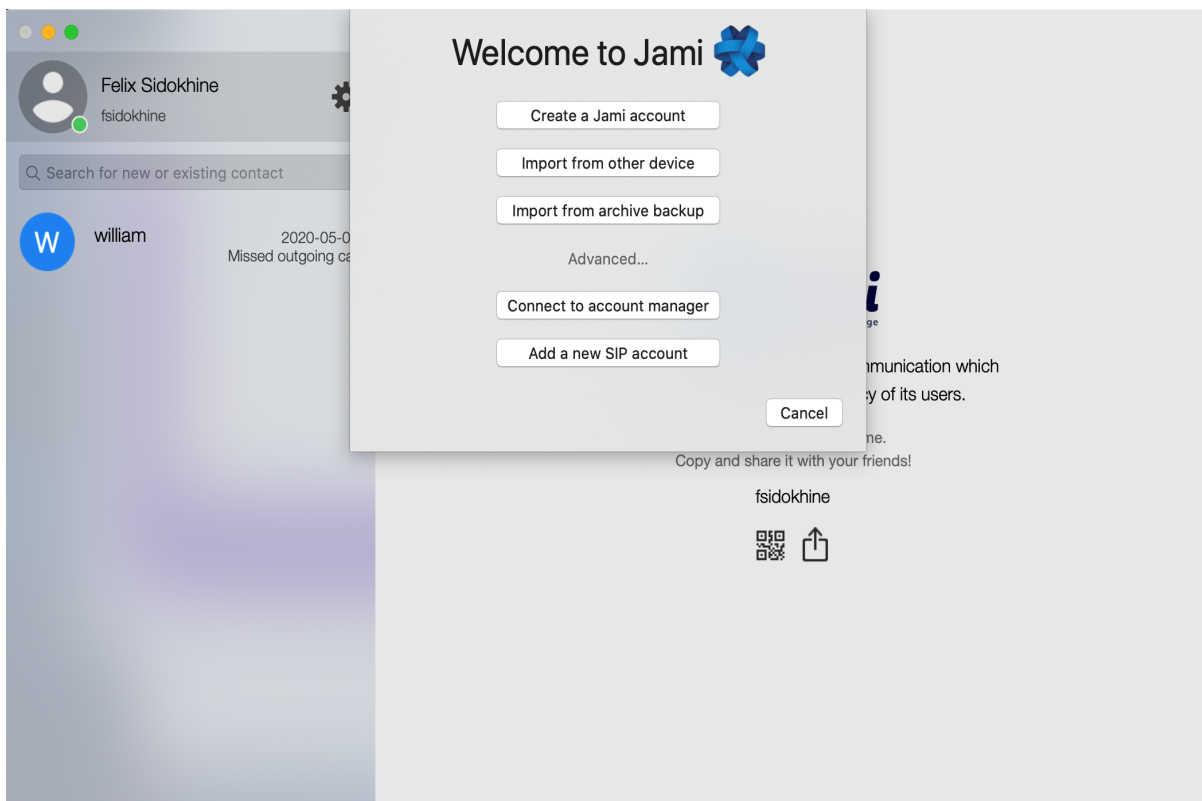
The server in this case would be the DNS address of your server and the username and password which correspond to your account. If you have configured the server with an LDAP/AD backend, it would be your LDAP/AD username and password.

Connecting using Mac OS

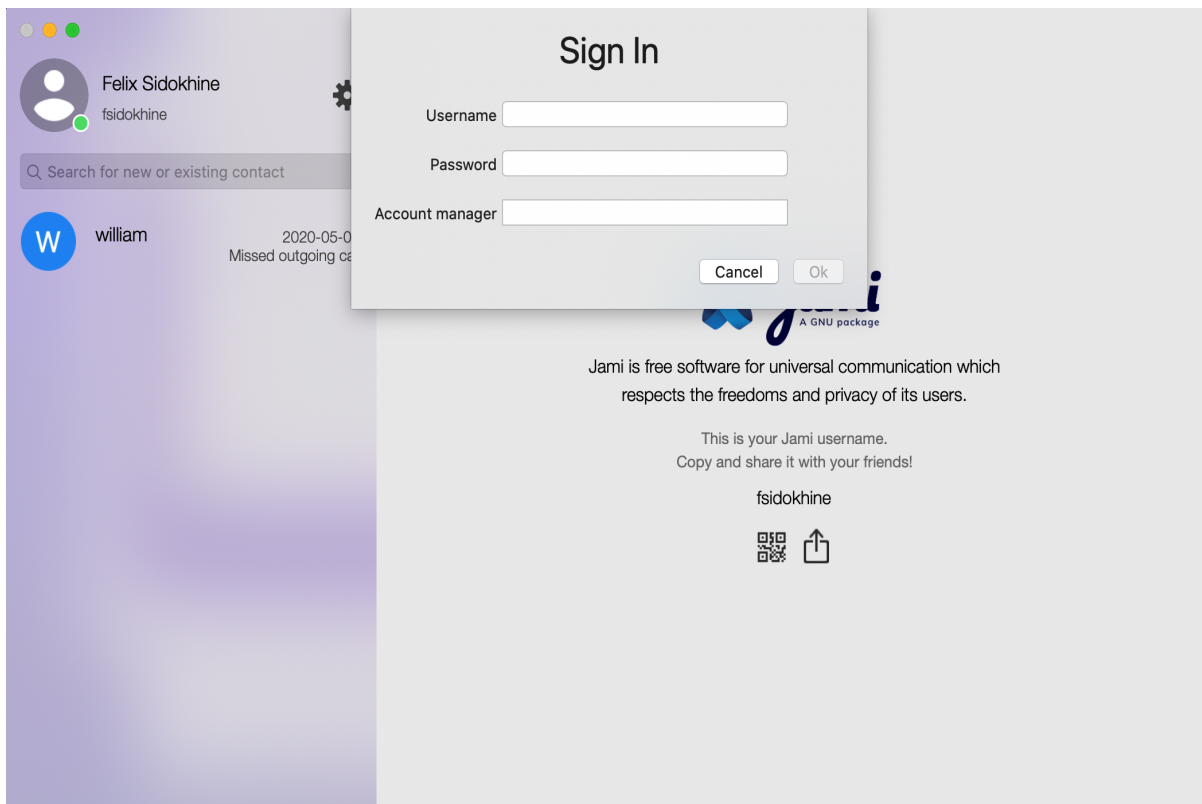
Upon opening Jami, you will be offered the following screen



Click on **Advanced** and additional options will appear:



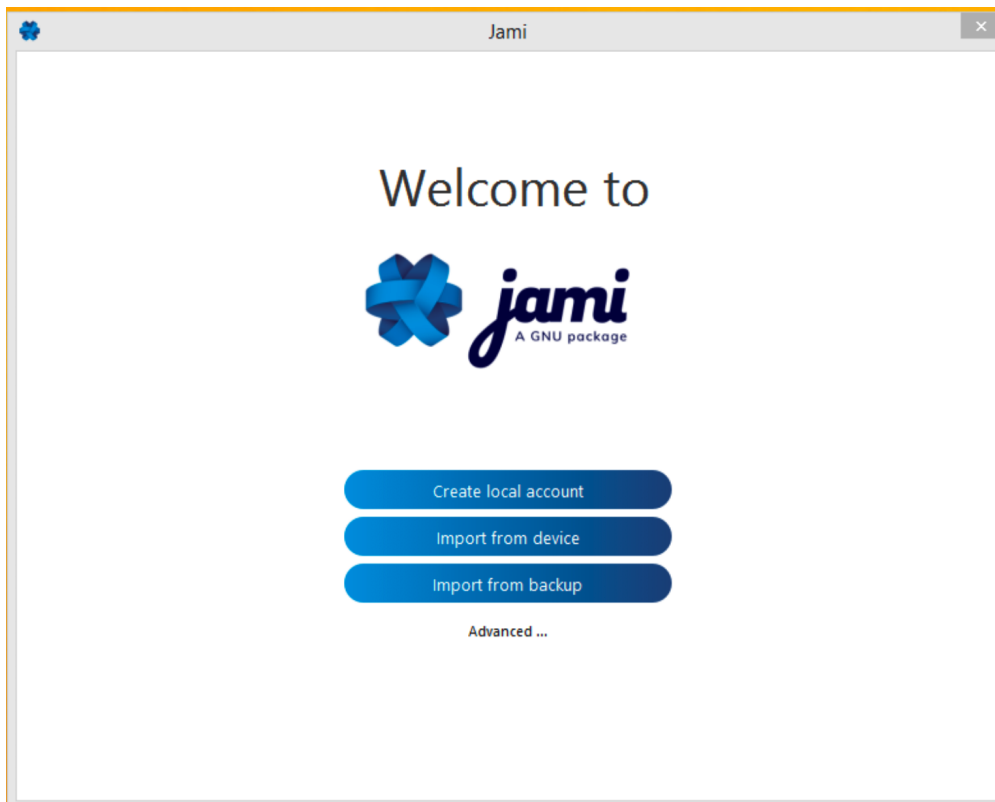
You should select the option "**CONNECT TO MANAGEMENT SERVER**" which will lead you to the following screen:



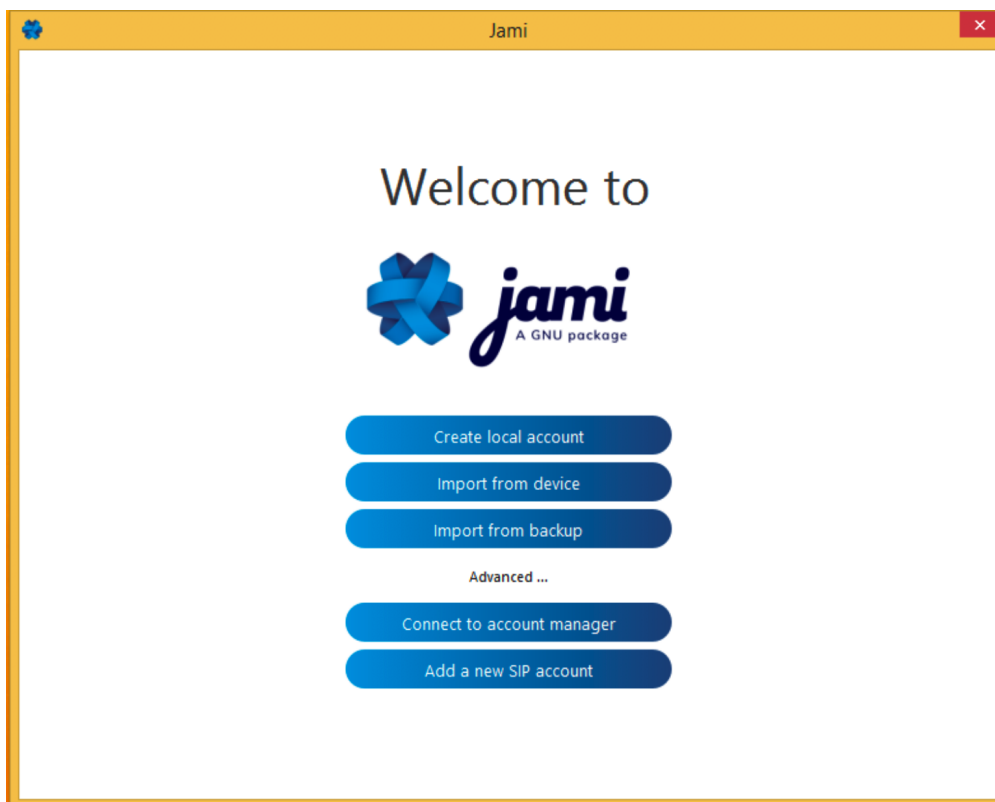
The Account manager in this case would be the DNS address of your server and the username and password which correspond to your account. If you have configured the server with an LDAP/AD backend, it would be your LDAP/AD username and password.

Connecting using Windows

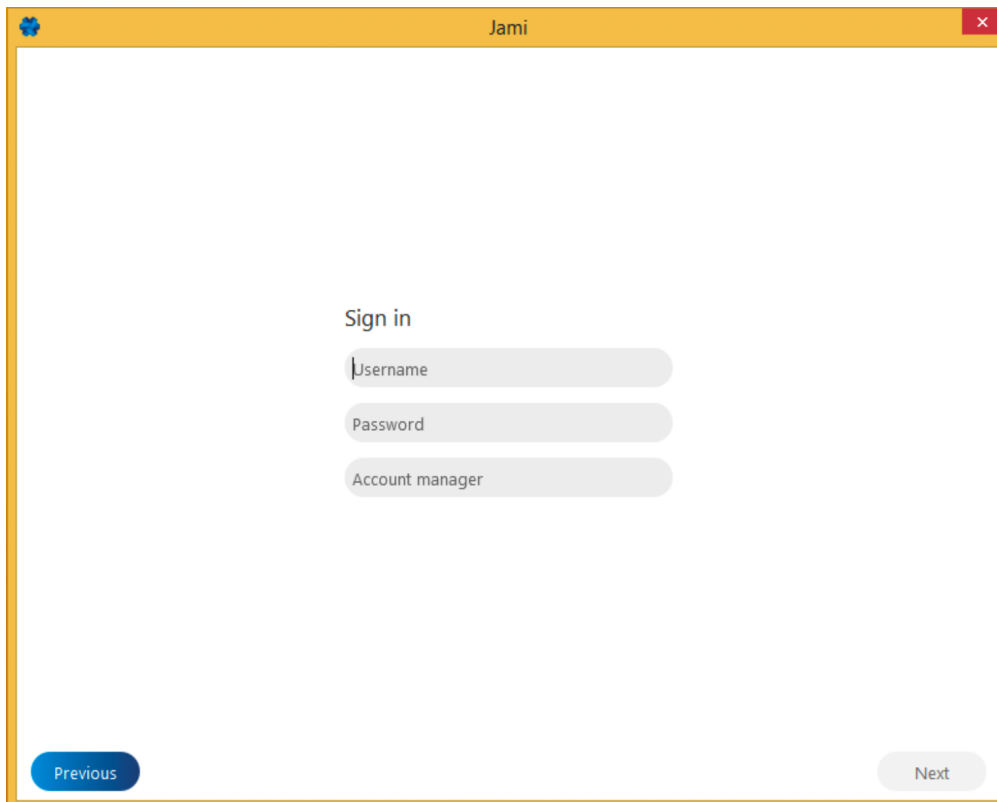
Upon opening Jami, you will be offered the following screen



Click on **Advanced** and additional options will appear:



You should select the option "**Connect to account manager**" which will lead you to the following screen:

A screenshot of a window titled "Jami" with a yellow header bar. The window contains a "Sign in" section with three input fields: "Username", "Password", and "Account manager". At the bottom left is a blue "Previous" button, and at the bottom right is a grey "Next" button.

Sign in

Username

Password

Account manager

Previous

Next

The Account manager 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.