Guide to Logins

For FastX 3.2

Introduction

Logging in is a fundamental part of FastX. Unauthenticated users are extremely limited in what they can do. For the most part, the only thing they are allowed to do is log in. Typically, web based applications are completely contained within the web server itself. There is never a need to change users, or run any process as a different user than the user running the website. This is not the case for FastX.

FastX is unique in that a user who is logged in to the web still may need to authenticate onto an actual system to start a new session. Otherwise all sessions would be run as the same user. This is a powerful feature of FastX; the ability to launch user sessions from the web.

Api Tokens

Api Tokens are the basis of logins. Api Tokens (or just tokens) can be used interchangeably with logins. FastX uses JSON Web Tokens (JWTs) to define the claims of a login. When authentication is successful, the web server issues a token that the user uses to authenticate subsequent requests. FastX contains a whitelist of all valid API Tokens. in the database so a token can be revoked by administrators (or the user himself). As a security measure, the API tokens in the database only contain the JWT claims, ensuring the database entries can only be used for verification, not to make API calls.

JSON Web Tokens

A JSON Web Token is a standard mechanism for passing signed data across systems. It is defined in RFC 7519. For more information see https://jwt.io/

Claims

The data in a JWT are called claims. The claims define the information of the JWT. In terms of FastX API Tokens, these claims are used as the data of the login. JWT claims are human readable and not encrypted. An API Token defines the following claims.

- sub -- username
- iss -- serverID
- iat -- issued at (in sec since epoc)
- exp -- expire time (in sec since epoc)
- aud -- "api" -- audience API (user login)
- jti -- unique identifier of the jwt. Defines the token in the database
- fastx/auth-method -- how the token was generated
    - web-ssh -- user called the ssh api login
    - web-openid -- user logged in with OpenId Connect
    - link -- user called the link program
    - link-daemon -- user/program called link with the --daemon option
    - ssh -- SSH client connection (via fastx-protocol command)
    - web-ssh-start -- link was generated via a start command (if no link-daemon was running on the system)
- fastx/client-ip -- (OPTIONAL) ip address where the auth-method came from
- fastx/hostname -- (OPTIONAL) Hostname of the issuer (for display purposes)
- fastx/socket -- (OPTIONAL) associated link socket
- fastx/daemon -- (OPTIONAL) this is a link daemon (only 1 per system. when a new daemon is added is added, all other daemons on the system are shut down)
- fastx/gateway-serverId -- (OPTIONAL) serverId of the gateway (if different from iss)
- fastx/gateway-hostname -- (OPTIONAL) Hostname of the gateway (if different from iss)
- fastx/node-link -- (OPTIONAL) did it spawn a node subprocess

Link Process

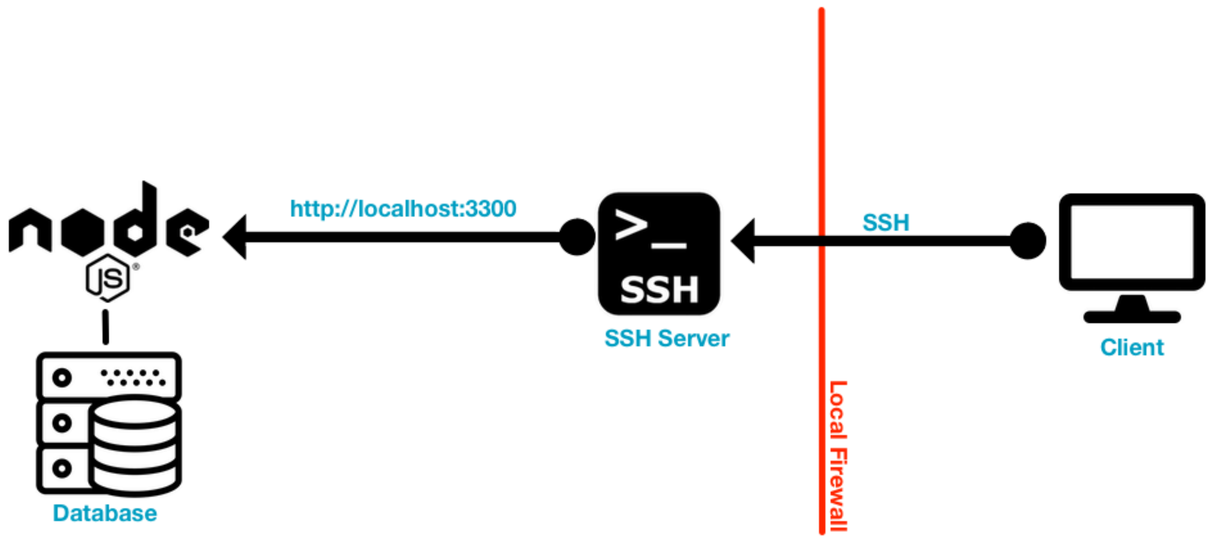
API Tokens are used extensively to pass information and authorize users to perform actions on the web server. However, the web server has no way to launch a session. In order to solve this issue, FastX creates a small background process called link as the user. The main purpose of link is to wait until a start command is called and then run the command. Depending on how you authenticate, a similar process called fastx-protocol may be used. For sake of simplicity, we will use the term link, or link process to refer to both instances.

API Tokens with the "fastx/socket" claim have an associated link process. This means that when a user issues an api call that needs to use the link process, this link process will be used. API Tokens that do not have the "fastx/socket" claim will attempt to use a link process that was marked as a Link Daemon to run actions as the user.

Authentication Methods

FastX offers multiple ways to authenticate to generate API Tokens.

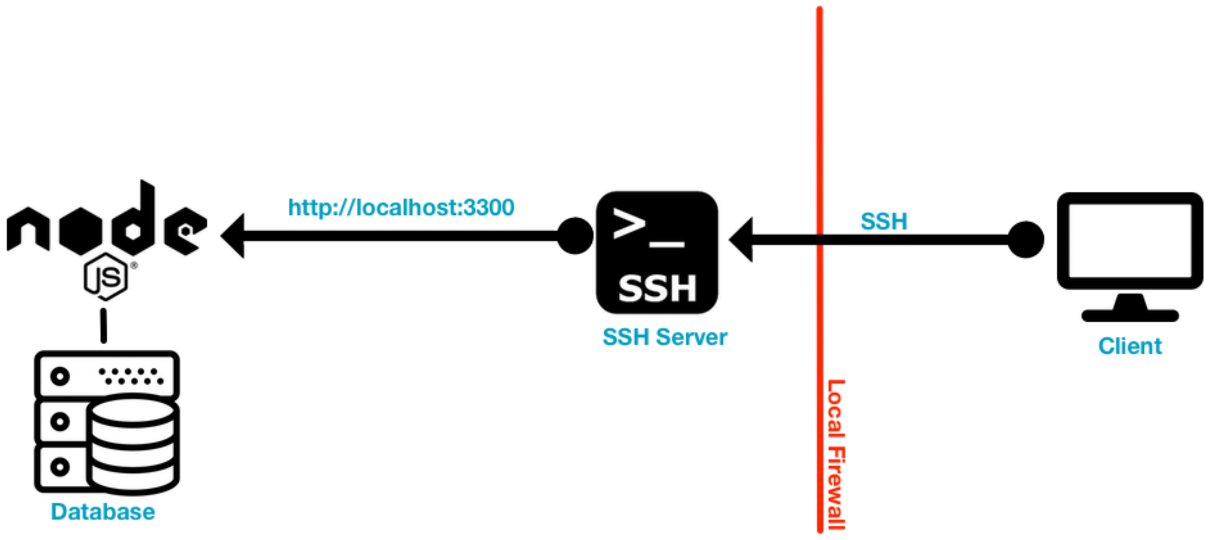
web-ssh



Web-ssh authentication is the default method to authenticate over the web. This is done for browser based authentication as well as Desktop Client authentication when connecting using the HTTPS method. When a user connects with his Linux username and password, FastX proxies the data over a secure SSL connection to the server. The server then creates a localhost ssh connection *ssh username@localhost* on the remote system. On a successful login, the newly created ssh connection will either create a new link-daemon if none exists, or use an existing link-daemon. After that the ssh connection will shut down leaving an API token to communicate with the server and a link-daemon to launch processes as the user.

web-ssh supports password, challenge response (duo, otp etc), public key authentication.

ssh



Ssh authentication is a special authentication method reserved exclusively for desktop clients. Ssh authentication first makes a direct ssh connection to the remote server. After the ssh connection successfully logs in, ssh then runs the fastx-protocol command which is used for communication between the ssh client and the FastX server. This connection is valid as long as the ssh connection stays up.

link

Link authentication is a general case of authentication when none has been specified, or when a user has run the /usr/lib/fastx/3/scripts/link from the command line. This is typically used to avoid having to authenticate to generate a token when you are already logged in to the system.

The link program is a process that runs in the background that allows the server to run processes as an authenticated user. Link is rarely used in this way by a user since he is already logged in to the system. The FastX server uses link extensively in the form of a link-daemon to start sessions.

link-daemon

A link daemon is a special case of Link that was started with the --daemon switch. Link daemons run in the background waiting for connections from the FastX web server. Any API token that does not have a "fastx/socket" claim will use the link-daemon as the way to launch sessions and perform other tasks as a user.

FastX attempts to limit the number of link daemons running per user to 1. When a new link daemon is initiated as a user, the FastX web server will attempt to shut down other link daemons.

web-openid

FastX supports OpenId Connect authentication. When a user authenticates with OpenId Connect, an API Token is generated. This token does not have a "fastx/socket" claim associated with it and must use a link-daemon in order to start sessions. Since OpenId Connect is purely web based, a user will either have to authenticate via SSH to start a session (see web-ssh-start), or the web server must be configured to allow sudo starts.

web-ssh-start

There are instances where a user may wish to start a session, but not have a link daemon running. Examples include: starting a session in a cluster, authenticating via OpenId Connect, the link daemon timing out or crashing. In these instances, FastX will attempt to authenticate via SSH before starting the session. The token that is created from this type of authentication is web-ssh-start

Proxy Authentication

Administrators who set a proxy authentication key can use the key to sign a JWT. This allows admins to bypass authentication and access the API without having to log in. A verified JWT is assumed to be valid. This makes it easy for admins to make API calls as a user, simplifying integration with other scripts. This can also be used to verify authentication using an auth method that FastX currently does not support.

In order to use proxy authentication.

1. Enable proxy Authentication by creating a secret key in *System* > *Authentication* > *Proxy*

Create a JWT with the following mandatory claims

```
{
    "sub": "USERNAME_TO_AUTHENTICATE_AS",
    "aud": "api",
    "iss": "proxy"
}
```

The "sub" claim is the user to authenticate as. -- FastX assumes this user exists. FastX does not check the database for this user and this key will not be added as a login.
The "aud" claim must be "api" -- Otherwise FastX will fail as an invalid API token
The "iss" claim must be "proxy" -- This tells FastX to verify against the proxy secret

The Link Process

Link Authentication

The FastX Web Server communicates with the link process via a unix domain socket. When the link process starts up it sends a request to the FastX Web Server's public https port with information on how to connect to the socket. The web server verifies the owner of the socket is the same user as the login specified in the authentication data. The web server then generates a JWT and sends it through the socket along with an initialization key.

The JWT is signed with the FastX Web Server's private key and the link verifies it with the FastX Web Server's public key. If the link process can not read the public key or the JWT signature is not valid the link process will refuse to execute commands.

Running Link From the Command Line

Link is the process that FastX uses to start sessions as a user. It is also used as a method to generate an API token when a user has already authenticated. This way a user does not have to log in again to access the API.

Run: */usr/lib/fastx/3/scripts/link* to generate an API token that can be used to communicate with the Web API

Link Daemons

Link daemons are special cases of links that are used to start sessions and perform other operations as a user from the web. Any token that does not have an associated "fastx/socket" claim will use the link daemon to start sessions. End users typically do not use Links in this way as starting a new link daemon may shut down your current link daemon.

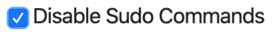
Run: */usr/lib/fastx/3/scripts/link --daemon=true* to launch a link daemon

Sudo Links

If a link daemon does not exist when a user needs one, the FastX Web Server will first attempt to start a link daemon using the sudo command. If this fails, the FastX Web Server will then attempt to launch a link daemon using SSH authentication.

To disable sudo attempts

1. Go to the admin System Section (gear icon)
2. Click on System > Settings
3. Check Disable Sudo Commands

   ✓ Disable Sudo Commands

4. Save

Desktop Client Logins

The Desktop Client has multiple ways of connecting to the remote system each with its own advantages and disadvantages. There is typically no performance penalty in choosing one connection method over the other.

HTTPS

HTTPS connections use the Web API to communicate directly with the FastX Web Server. The web server MUST be running to connect via HTTPS. The Desktop Client sends the authentication data over an encrypted SSL channel at which point the server tries to authenticate to its local SSH daemon ie: *ssh username@localhost*. This is the same method the Browser Client uses to authenticate.

HTTPS connections can use the server's load balancer to determine where the authentication can take place

SSH

The Desktop Client makes a standard SSH connection and authenticates BEFORE any interaction with FastX. Upon successful authentication, the ssh connection launches the fastx-protocol process which works as a communication channel between the client and server.

SSH connections run over a standard SSH channel which may be beneficial in secure environments.

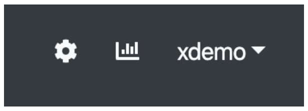
SSH connections are always direct point to point connections meaning that it does not use the FastX load balancer for logins.

SSH connections do not require the web server to be running.

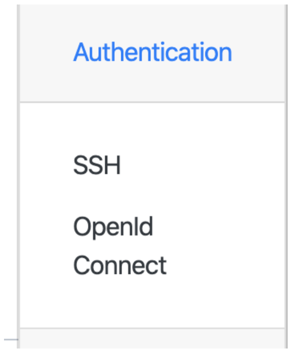
Load Balancing

SSH Logins can be load balanced across a cluster allowing even distribution of users across a cluster. This can also avoid double authentication when starting sessions if the user only starts sessions on his logged in server.

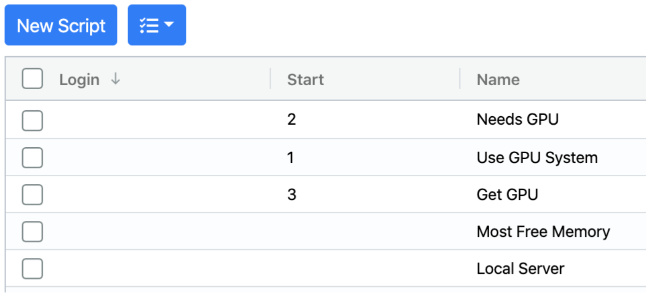
1. Click on the System Section (gear icon)



2. Click on Authentication > SSH



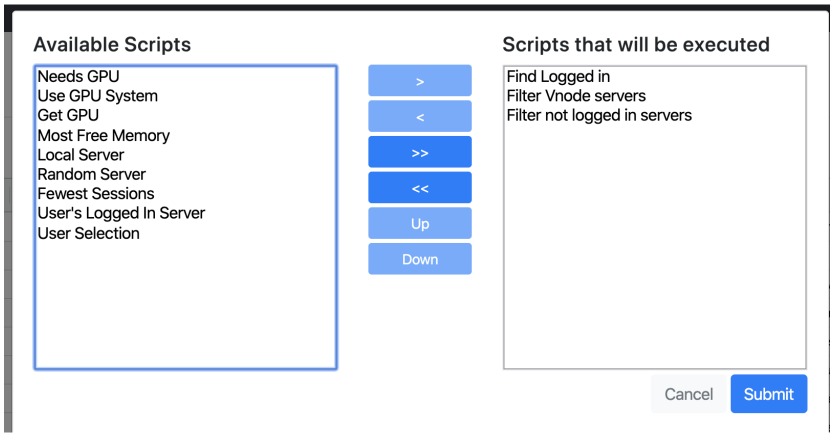
3. Multiple Predefined scripts are shipped with FastX, or an administrator can create his own script by clicking New Script



	Login	Start	Name
☐		2	Needs GPU
☐		1	Use GPU System
☐		3	Get GPU
☐			Most Free Memory
☐			Local Server

4. To Set your scripts, click on the Actions Icon and set the script

5. Then choose the order of the scripts you want executed and submit

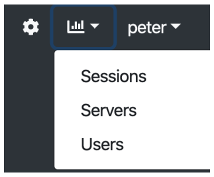


Managing Logins

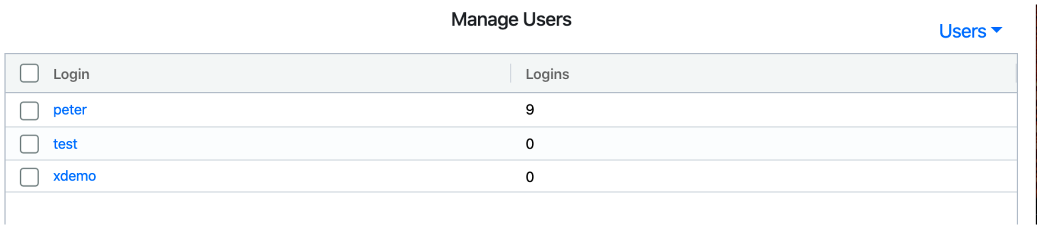
For Managers

Managers (with the login permission) can view and terminate logins from all users

1. Click on the Manage Section (graph icon) and choose Users



2. You will see an overview of all the users who have ever logged into your cluster (or standalone system)

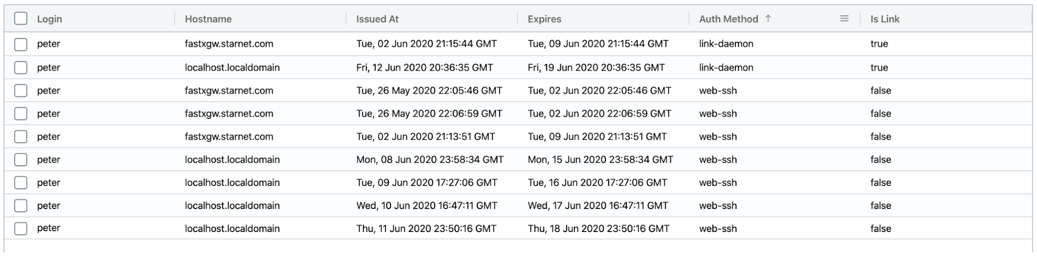


Manage Users

Users

Login	Logins
peter	9
test	0
xdemo	0

3. Click on a user to get detailed information of the logins. From here you can delete any login.



Login	Hostname	Issued At	Expires	Auth Method	Is Link
peter	fastxgw.starnet.com	Tue, 02 Jun 2020 21:15:44 GMT	Tue, 09 Jun 2020 21:15:44 GMT	link-daemon	true
peter	localhost.localdomain	Fri, 12 Jun 2020 20:36:35 GMT	Fri, 19 Jun 2020 20:36:35 GMT	link-daemon	true
peter	fastxgw.starnet.com	Tue, 26 May 2020 22:05:46 GMT	Tue, 02 Jun 2020 22:05:46 GMT	web-ssh	false
peter	fastxgw.starnet.com	Tue, 26 May 2020 22:06:59 GMT	Tue, 02 Jun 2020 22:06:59 GMT	web-ssh	false
peter	fastxgw.starnet.com	Tue, 02 Jun 2020 21:13:51 GMT	Tue, 09 Jun 2020 21:13:51 GMT	web-ssh	false
peter	localhost.localdomain	Mon, 08 Jun 2020 23:58:34 GMT	Mon, 15 Jun 2020 23:58:34 GMT	web-ssh	false
peter	localhost.localdomain	Tue, 09 Jun 2020 17:27:06 GMT	Tue, 16 Jun 2020 17:27:06 GMT	web-ssh	false
peter	localhost.localdomain	Wed, 10 Jun 2020 16:47:11 GMT	Wed, 17 Jun 2020 16:47:11 GMT	web-ssh	false
peter	localhost.localdomain	Thu, 11 Jun 2020 23:50:16 GMT	Thu, 18 Jun 2020 23:50:16 GMT	web-ssh	false

For Users

Users can manage their own logins by clicking on their name, choosing profile and selecting logins.

Logging Out

Logins are defined by their API Tokens. Only API Tokens that are in the database are considered valid (An API Token is in the database if its "jti" claim is in the database). Deleting an API token from the database will effectively log out that user's token.

If the API token has the "fastx/socket" claim, the FastX Web Server will also attempt to shut down the associated link process related to the token.

Shutting Down Link Daemons

Admins who do not want to keep link daemons around after logout should do the following

5. Go to the admin System Section (gear icon)
6. Click on Users > Logins
7. Check Shutdown Link Daemon on Logout

   ✓ Shutdown Link Daemon on Logout

8. Save

Token Expiration

All API Tokens expire which invalidates the token. Associated links will terminate themselves when the token has expired.

StarNet Communications Corp.
4677 Old Ironside Drive, Suite 210, Santa Clara, California 95054-1825 • USA
408.739.0881 • 408.739.0936 • sales@starnet.com • www.starnet.com