

NDEx Installation Instructions

Last updated: July 26th, 2018

Overview

These step-by step instructions will guide you through a complete de novo installation of NDEx v2.0.0 or higher.

Step 1 – SYSTEM SETUP

1a) Java 8

Make sure Java 8 is installed in your system. For instructions on how to install Java 8, click here (<http://www.java.com/en/download/manual.jsp>).

1b) Apache

Install the Apache HTTP server **version 2.4**.

1c) PostgreSQL

Install the PostgreSQL server **version 9.5**. The PostgreSQL server can be install on the same machine where you run the NDEx server, or can be installed on a separate machine.

1d) Install Python (if not already installed on your server)

NDEx v2.0.0 or higher **requires Python 2.7**. The default NDEx installation requires Python interpreter at `/usr/bin/python`. You also need to install the following Python modules: **gevent**, **gevent_websocket**, **bottle** and **pysolr**.

1e) Create the ndex user account

```
# -M, --no-create-home do not create the user's home directory
```

```
# -r, --system create a system account
```

```
# -s, --shell SHELL login shell of the new account (/bin/false = no login)
```

```
# -U, --user-group create a group with the same name as the user
```

```
sudo useradd -M -r -s /bin/false -U ndex
```

***** IMPORTANT NOTE *****

In the following instructions, the ndex account is used to run tomcat server (and thereby the NDEx REST server) and all files are configured with the ndex user as owner. The tomcat7 start and stop scripts automatically use the ndex user. In all other situations, **it is necessary** to assume the role of the ndex user with "sudo su - ndex".

Step 2 – DOWNLOAD SOFTWARE

The NDEx bundle is a compressed archive and can be downloaded from our FTP server.

2a) Obtain the latest NDEx bundle from [ftp.ndexbio.org](ftp://ftp.ndexbio.org) (`ftp://ftp.ndexbio.org`). In this example, we use the ndex-2.3.1 archive. The archive can be downloaded from the command line with **wget:**

```
cd /opt
```

```
sudo wget ftp://ftp.ndexbio.org/NDEx-v2.3.1/ndex-2.3.1.tar.gz
```

2b) Now extract the ndex-2.3.1.tar.gz archive using the commands below:

```
cd /opt
```

```
sudo gzip -d ndex-2.3.1.tar.gz
```

```
sudo tar xvf ndex-2.3.1.tar
```

```
sudo chown -R ndex:ndex ndex
```

The archive will be extracted to the ndex directory regardless of the version you have downloaded. Symbolic links to omcat and Solr will also be created automatically. The last command line is required to change ownership of the newly created ndex directory.

After extraction has completed, the directory should look like:

```
/opt
/ndex
/apache-tomcat-x.x.xx
/bin
/conf
/dbbackups
/exported-networks
/ndex-webapp
/resources
/solr -> solr-5.4.1
/solr-5.4.1
/tomcat -> apache-tomcat-x.x.xx
/uploaded-networks
/workspace
```

Step 3 – CONFIGURATION

3a) Configuring the Apache web server

The Apache web server must be configured to:

- Serve the NDEx website
- Make the NDEx REST server, running as a Tomcat webapp, available at a standard, convenient URL
- (This is done by establishing a reverse proxy, an "alias" for the NDEx server's address)

Details:

- The Tomcat main page is served at host:8080
- Tomcat makes the REST server webapp available at host:8080/ndexbio-rest
- In the typical configuration, the ndex web ui is served by Apache on the same server
- The document root is changed to /opt/ndex/ndex-webapp
- (The files in /opt/ndex/ndex-webapp are from the project ndex-webapp)
- To conveniently use the REST server from the ndex web ui we setup a proxy so that it will be available as a "folder" of the website.
- For example, if the website is deployed at **www.ndexbio.org**, the REST server will be at **www.ndexbio.org/v2**

The configuration is accomplished by adding an additional configuration file that Apache will read after loading its main configuration. This file must be added to the Apache installation. The location of the file depends on the version of Unix that is being used.

CentOS

In CentOS (and RedHat), changes to the Apache server configuration are accomplished by adding a new config file called **ndex.conf** under the **/etc/httpd/conf.d** directory. A typical setting in the ndex.conf file would be like this:

```
<VirtualHost *:80>
ServerAdmin support@ndexbio.org

DocumentRoot /opt/ndex/ndex-webapp

<Directory />
Options FollowSymLinks

AllowOverride None

</Directory>

<Directory /opt/ndex/ndex-webapp>
Options Indexes FollowSymLinks MultiViews

AllowOverride None

Order allow,deny

allow from all

</Directory>

<FilesMatch "\.(?i:xgml|xbel)$">
Header set Content-Disposition attachment

</FilesMatch>

ProxyPass /rest/ http://localhost:8080/ndexbio-rest/
ProxyPassReverse /rest/ http://localhost:8080/ndexbio-rest/

ProxyPass /v2/ http://localhost:8080/ndexbio-rest/v2/ timeout=3000
ProxyPassReverse /v2/ http://localhost:8080/ndexbio-rest/v2/

ProxyPass /V2/ http://localhost:8080/ndexbio-rest/v2/ timeout=3000
ProxyPassReverse /V2/ http://localhost:8080/ndexbio-rest/v2/

</VirtualHost>
```

Ubuntu

In Ubuntu, changes to the Apache server configuration are accomplished by adding a new config file **ndex.conf** under the `/etc/apache2/sites-enabled` directory. A typical setting in the `ndex.conf` file would be like this:

```

<VirtualHost *:80>
ServerAdmin support@ndexbio.org

DocumentRoot /opt/ndex/ndex-webapp

<Directory />
Options FollowSymLinks

AllowOverride None

</Directory>

<Directory /opt/ndex/ndex-webapp>
Options Indexes FollowSymLinks MultiViews

AllowOverride None

Order allow,deny

allow from all

</Directory>

<FilesMatch "\.(?i:xgml|xbel)$">
Header set Content-Disposition attachment

</FilesMatch>

ErrorLog ${APACHE_LOG_DIR}/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.

LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined

ProxyPass /rest/ http://localhost:8080/ndexbio-rest/

ProxyPassReverse /rest/ http://localhost:8080/ndexbio-rest/

ProxyPass /v2/ http://localhost:8080/ndexbio-rest/v2/ timeout=3000

ProxyPassReverse /v2/ http://localhost:8080/ndexbio-rest/v2/

ProxyPass /V2/ http://localhost:8080/ndexbio-rest/v2/ timeout=3000

ProxyPassReverse /V2/ http://localhost:8080/ndexbio-rest/v2/

</VirtualHost>

```

3b) Initialize the PostgreSQL database

The NDEX 2.0 server uses PostgreSQL server as a backend database. The PostgreSQL database needs to be initialized and started before you start the NDEX 2.0 server. You can use this command to create a user and a database in your PostgreSQL server:

```
-bash-4.2$ psql
psql (9.5.4)
Type "help" for help.

postgres=#

create role ndexserver LOGIN password 'my_password' NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE NOREPLICATI
ON;

ALTER ROLE ndexserver
SET search_path = core, "$user", public;

CREATE DATABASE ndex
WITH OWNER = ndexserver
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
CONNECTION LIMIT = -1;

\q
```

After the database and user are created. You can create the schema using the file scripts/ndex_db_schema.sql. The command can be something like this:

```
-bash-4.2$ psql ndex <~/ndex_db_schema.sql
```

***** NOTE:** you might need to modify the pg_hba.conf file to allow connections from NDEx server. For example, you can add the following line to allow the ndexserver user to connect from the same server where the Postgres server is installed:

```
local ndex ndexserver md5
```

3c) Changing NDEx server properties

The NDEx server configuration file is called **ndex.properties** and can be found under directory /opt/ndex/conf.

!!! The default values of the following properties should never be modified !!!

```
NdexSystemUser=ndexadministrator
NdexSystemUserPassword=admin888
NdexSystemUserEmail=support2@ndexbio.org
```

1) Change the **HostURI** property. You need to set its value to the host name of your machine with the http prefix.

For example, if you are installing NDEx to a machine named *myserver.somedomain.com*, the HostURI value should be set to: *HostURI=http://myserver.somedomain.com*

2) The **SMTP-XXXX** properties need to be updated only if you want to allow users to update their passwords.

3) To enable **LDAP Server Authentication**, you will need to edit the ndex.properties configuration following properties:

USE_AD_AUTHENTICATION= This should be set to "true" if you want to turn on LDAP authentication. Default value is *false*.

AD_USE_SSL= Set to true if you want to use SSL with LDAP. Default value is *false*.

PROP_LDAP_URL= This property specifies the URL of your LDAP server. For example, it can be *ldap://dir.mycompany.com:389* for non-secured server or *ldaps://dir.mycompany.com:636* for secured server.

AUTHENTICATED_USER_ONLY= The NDEx server will run in "Authenticated user only" mode when this value is set to true. In this mode, all API functions require user authentication except: */admin/status*, */user/authenticate* and *create user*. Default value is *false*.

KEYSTORE_PATH= This is the path of Java keystore in your JVM. This value is required when "AD_USE_SSL" is set to *true*.

JAVA_KEYSTORE_PASSWD= The password of your Java keystore if you have a password setup for it.

AD_CTX_PRINCIPLE= The string pattern to use when setting the SECURITY_PRINCIPAL context in the LDAP authentication. For example, if you set this value to "NA\\%%USER_NAME%%", the server will append string "NA\\" to your user name and use it to set the Context. SECURITY_PRINCIPAL value in the LDAP search. %%USER_NAME%%" is a reserved word in NDEX LDAP setting, it will be replaced by the user's user name in LDAP queries.

AD_SEARCH_FILTER= The string pattern to be used in the LDAP search. For example it can be something like: *(&(objectclass=user)(cn=%%USER_NAME%%))*.

AD_SEARCH_BASE= (Optional) This property defines the search base parameters: for example, if you want to search in the domain *my.company1.com*, you can define the property as: *AD_SEARCH_BASE=DC=my,DC=company,DC=com*. If you don't define this property, no search base will be used in the LDAP authentication.

AD_NDEX= (Optional) If this property is defined, only the users in the declared group will be allowed to create accounts and use the NDEx server.

AD_DELEGATED_ACCOUNT= (Optional) In some use cases. The authentication has 2 steps. 1) Using a generic account to connect to LDAP server and run a query on the LDAP server on the accountName to get a fully qualified name of that user. 2) Use the fully qualified name to authenticate the user. The username and password of the generic account can be defined in this parameter and AD_DELEGATED_ACCOUNT_PASSWORD property. No generic account is used if this parameter is not defined.

When this parameter is defined, AD_DELEGATED_ACCOUNT_PASSWORD becomes a required parameter.

AD_DELEGATED_ACCOUNT_PASSWORD= (Optional) Required when AD_DELEGATED_ACCOUNT is defined.

AD_CREATE_USER_AUTOMATICALLY= If AD authentication is turned on and this parameter is set to true, when a user logs in successfully for the first time using LDAP, the NDEx server will automatically create an NDEx account for that user. The NDEx server uses this user's "givenName", "sn" and "mail" attributes in the AD record as his firstName, lastName and emailAddress when creating the NDEx account.

AD_CTX_PRINCIPLE2= (Optional) The NDEx administrator can set this parameter in ndex.properties to enable the use of a second domain to search in the LDAP server.

4) The Log-Level parameter controls how much log information is written to the *ndex.log* file located in the */opt/ndex/tomcat/logs* directory. Possible values are **info**, **error** and **off**. The default value is **info**: in this mode, a log entry is created at the beginning and end of every API call on the server that also includes the error (exception) information. Setting Log-Level to **error** will only log exceptions. To disable logging, set Log-Level to **off**. IMPORTANT: after changing the Log-Level value, you need to restart your server for the new setting to take effect.

5) NeighborhoodQueryURL The Root URL of the Neighborhood Query Endpoint. The default value is *http://localhost:8284/query/v1/network/*.

6) The NDEx v2.0 Server supports email verification upon account creation. The configuration parameter is **VERIFY_NEWUSER_BY_EMAIL**. The default value is *false*. When it is set to *true*, new accounts created on the server will be required to verify the email address used for registration. The createUser function has been modified to implement the first part of this feature. When user creates an account and the server requires email verification, the object returned from this function will not have a UUID value for the user, and the server will send a verification email to the user. Here is an example:

```
Dear <_First name Last name_>

Thank you for registering an NDEx account.

Please click the link below to confirm your email address and start using NDEx now! You can also copy and paste the link in a new browser window.

>>LINK HERE>>

This is an automated message, please do not respond to this email. If you need help, contact us by email: support@ndexbio.org

Best Regards,

The NDEx team
```

A new rest API function implements the acceptance of the verification code and activation of the account.

@GET

@PermitAll

@Path("/{userId}/verify/{verificationCode}")

The NDEx Web UI has been modified to redirect the new user to a verification page instead of their homepage, if verification is enabled. On that page the user will be informed to check his email and click the link in the confirmation email to validate his address. The link will make an API call to perform the verification; if the verification succeeds, the API will return a User object and the new user (with an activated account) will now be able to login to his newly created NDEx account.

7) Configure the connection parameter to PostgreSQL database. These 3 parameters need to be set in the configuration file:

```
NdexDBURL=jdbc:postgresql://localhost:5432/ndex
NdexDBUsername=ndexserver
NdexDBDBPassword=ndex
```

8) Set these parameters if you want to enable the Google OAuth feature on the server:

```
USE_GOOGLE_AUTHENTICATION=true
GOOGLE_OAUTH_CLIENT_ID=xxxxx.apps.googleusercontent.com
```

You can get a Google OAUTH Client Id by registering your server with a Google developer account through the Developers Console (<http://console.developers.google.com/>).

3d) Changing web app properties

The NDEx web-app configuration file (*ndex-webapp-config.js*) can be found under directory `/opt/ndex/ndex-webapp`. Here is a list of the properties that can currently be configured:

welcome: Consists of three parts: header, linkToReleaseDocs and message. header specifies the welcome message that should be displayed on the landing page of the web app. Default value is: "NDEx Web App deployed at My Company". linkToReleaseDocs points to the release notes of the NDEx version specified by the server. The message field has no default value but you can add any text to describe your NDEx installation, policy, rules, etc.

logoLink: This property allows users to customize the URL linked to the NDEx logo in the top left corner of the web UI. The default link is: <http://www.home.ndexbio.org> (<http://www.home.ndexbio.org/>). Users can also define a customized "warning" message to display when clicking on the logo and can decide whether to show the warning or not.

newsLink, aboutLink, documentationLink, reportBugLink and **contactUsLink:** These options allow users to fully customize the look and behavior of the links in the top menu bar. For each link, users can specify:

- label (text)
- href (URL)
- warning (text)

- showWarning (true or false)

messages: This option allows to specify a custom message to be displayed when the server is unavailable. The "serverDown" property can be defined using HTML strings to display text, images or a combination of both.

requireAuthentication: This option specifies whether authentication is required to use the web app. If authentication is required (property set to "true"), anonymous searches of the NDEx server through the NDEx web app are disabled: this is achieved by hiding the search bar. Default value is: *false*.

signIn: this section has 4 configurable elements. *Footer* and *header* to customize the text in the upper and lower parts of the sign in box; *showForgotPassword* and *showSignUp* allow to control whether the "Forgot Password" and "Sign Up" links are displayed or not. Default value for these 2 properties is *true*.

searchDocLink: At the bottom of Search Examples drop down menu on the NDEx landing page, defines the URL where the "NDEx search" user manual can be found.

featuredCollections: Allow to customize access to selected network collections directly from the NDEx landing page.

refreshIntervalInSeconds: Integer number specifying time interval in seconds for automatic refresh of My Account page content for logged in users. Default value is 0 (no auto-refresh).

ndexServerUri: Specifies the ndex server in use. Currently, NDEx only supports the *http protocol*. *Https* support will be added in future releases.

networkQueryEdgeLimit: Specifies the maximum number of edges that may be returned by a query before the query service attempts to re-run in background and save the result directly to the user's account. For anonymous access, when the result of a neighborhood query exceeds the *networkQueryEdgeLimit*, users are prompted to log in so that the result can be saved to their account. Default recommended value is: *50000*.

idleTime: Specifies the amount of time (in seconds) after which the user is automatically logged out for inactivity. Default value is: *3600*

uploadSizeLimit: Specifies the maximum file size (in Mb) that can be uploaded using the web UI. Default value is: *none*, that means there is no size limit.

googleClientId: Specifies the Google Client Id of the NDEx server this webapp is connecting to.

openInCytoscapeEdgeThresholdWarning: When opening a network in Cytoscaspe, users will be warned about possible performance issues if the network is larger than the threshold specified. Default value for this property is *100000*. Setting the value to *0* disables the warning.

Note - The following configuration parameters are no longer supported in this version:
NETWORK_POST_ELEMENT_LIMIT

3e) Starting and stopping Apache

Now that you have finished configuring Apache, you may start it so that the front-end of your NDEx server runs. Overall, for your NDEx server to run properly, both Apache and Tomcat must be running.

CentOS

Start

```
sudo /sbin/service httpd start
```

Stop

```
>sudo /sbin/service httpd stop
```

Restart

```
sudo /sbin/service httpd restart
```

Ubuntu

Start

```
sudo /etc/init.d/apache2 start
```

Stop

```
sudo /etc/init.d/apache2 start
```

Restart

```
sudo /etc/init.d/apache2 start
```

3f) Managing Tomcat as a service (OPTIONAL)

Scripts in the `/etc/init.d` directory may be used to run processes as services. There is a service script called `tomcat7` in `/opt/ndex/bin` and you can copy it to `/etc/init.d` and register tomcat as a service:

```
sudo cp /opt/ndex/bin/tomcat7 /etc/init.d
```

To register the service in CentOS, use this command:

```
chkconfig --add tomcat7
```

To register the service in Ubuntu, use this command:

```
update-rc.d tomcat7 defaults
```

Step 4 – START THE NDEX-REST SERVER

4a) Starting Solr

NDEX v2.0 has **Solr 6.5.1** as a component in the server bundle. The Solr service needs to be started before the NDEX Tomcat server is started. To start the Solr service, use the following commands (assuming that the NDEX bundle is installed under directory `/opt/ndex`):

```
cd /opt/ndex/solr
bin/solr start -m 32g
```

4b) Starting (and stopping) the Tomcat server

If you have registered Tomcat as a service (see OPTIONAL Step 3f), you can start and stop the service using these commands:

```
sudo service tomcat7 start
sudo service tomcat7 stop
```

If you decided not to register Tomcat as a service, you can start and stop the service with its standard scripts under `/opt/ndex/tomcat/bin`

```
cd /opt/ndex/tomcat/bin
sudo su - ndex
bash startup.sh
bash shutdown.sh
```

***** NOTE:** if you are having any trouble getting Tomcat or NDEx configured, its a good idea to launch it "manually" without detaching so that you can see any errors:

```
sudo su - ndex
bash catalina.sh run
```

4c) Start the Query Service

Go to the directory `query_engine` and run the script "run.sh" to start the neighborhood query engine.

4d) Proxy Issues

If after completing these steps the front-end of your NDEx server does not seem to be talking to the back-end, it may be because your security settings are preventing your proxy settings from going into effect. If you believe this may be the case, please see your local system administrator.

CONGRATULATIONS !!! You have successfully installed the NDEx REST server and web application user interface.