1. OpenAM 2
1.1 Acquire OpenAM Packages 2
1.2 Agent Installation 3
1.3 Application Configuration 8
1.4 OpenAM Configuration 9
1.4.1 OpenAM User/Pass Encryption 11
1.5 OpenAM Installation 11
1.6 Server Setup 12

OpenAM

Home for OpenAM configuration and information.

> Current OpenAM versions NICS6 uses:
>
> - **OpenAM 12.0.0**
> - **Web Agent 3.3.4**

Steps for setting up OpenAM for NICS, in order:

1. Server Setup
2. Acquire OpenAM Packages
3. OpenAM Installation
4. OpenAM Configuration
5. Agent Installation
6. Application Configuration

These steps assume your VMs are already set up with the basic NICS requirements, such as Java and Apache being installed.

Acquire OpenAM Packages

Guide to downloading the full OpenAM release

Acquire the main OpenAM package

1. If you do not have one already, you will need a Backstage login at Forgerock to get the latest OpenAM enterprise downloads
2. Go to OpenAM 12.0.0 Enterprise. It should have pre-selected for you the OpenAM Enterprise 12.0.0 zip download
3. If you have a Backstage account, and are logged in, you should see a Download button. If not, it'll be labeled "Sign in to Download". Click that
4. If it's asking you to sign in and you don't have an account, follow their instructions to create one. Once you've created your account, navigate back to the link in Step #2, and download the zip file
5. Once you've acquired the zip file, move it to the server on which you wish to configure OpenAM
6. For the purposes of this guide, we'll assume you placed it in /opt
7. Unzip the zip file:

```
> unzip OpenAM-12.0.0.zip
...
```

8. You should now have an /opt/openam directory. CD into the directory. You should see the contents below:

```
> ls
ClientSDK-12.0.0.jar             Fedlet-12.0.0.zip        legal-notices
OpenAM-ServerOnly-12.0.0.war
ExampleClientSDK-CLI-12.0.0.zip  IDPDiscovery-12.0.0.war  OpenAM-12.0.0.war
SSOAdminTools-12.0.0.zip
ExampleClientSDK-WAR-12.0.0.war  ldif
OpenAM-DistAuth-12.0.0.war  SSOConfiguratorTools-12.0.0.zip
>
```

9. Files of note:
    a. OpenAM-12.0.0.war - This is what we deploy to Tomcat. Just remember we change the name of the file to openam.war when deploying to Tomcat.
    b. SSOAdminTools-12.0.0.zip - We need this package to encrypt credentials. It's also useful for making other configuration changes, backups, etc.

Acquire the Web Agent package

1. Go to OpenAM Web Agent 3.3.4 Apache 2.4 Linux
2. Click Download (or follow steps above to Sign in to Download and create an account if you don't have one)
3. Move the downloaded file to your NICS Data VM

Agent Installation

Guide for setting up an OpenAM Web Agent. See OpenAM's documentation as a reference: https://backstage.forgerock.com/#!/docs/openam-policy-agents/3.3.0/web-install-guide/

The below guide shows you how to set up the Agent for the API. You can also set one up in front of your Mapserver, if you have one.

> Note: Even though we're using 3.3.4, in many cases the OpenAM documentation wasn't changed since 3.3.0. Just be sure you've downloaded version 3.3.4 of the Web Policy Agent.

Table of Contents

- Prerequisites
- Setup Web Agent
- Add Agent to OpenAM
- Configure Policies

Prerequisites

- Acquire OpenAM Packages

Setup Web Agent

For the purposes of this guide, we'll assume the you placed the *Apache-v2.4-Linux-64-Agent-3.3.4.zip* file in the /opt directory on the Data VM.

1. Unzip the archive:

```
> cd /opt
> unzip Apache-v2.4-Linux-64-Agent-3.3.4.zip
Archive:  Apache-v2.4-Linux-64-Agent-3.3.4.zip
   creating: web_agents/
   ...
>
```

   You now have a 'web_agents' directory
2. cd into the apache agent directory:

```
> cd web_agents/apache24_agent
> ls -l
drwxr-xr-x 2 user user 4096 Jan 15  2015 bin
-rw-r--r-- 1 user user 8770 Jan 15  2015 binary-license.txt
drwxr-xr-x 2 user user 4096 Jan 15  2015 config
drwxr-xr-x 2 user user 4096 Jan 15  2015 data
drwxr-xr-x 2 user user 4096 Jan 15  2015 etc
drwxr-xr-x 2 user user 4096 Jan 15  2015 installer-logs
drwxr-xr-x 2 user user 4096 Jan 15  2015 lib
-rw-r--r-- 1 user user 8770 Jan 15  2015 license.txt
drwxr-xr-x 2 user user 4096 Jan 15  2015 locale
-rw-r--r-- 1 user user 5797 Jan 15  2015 README
>
```

> If you've previously installed an agent on this machine, there'a a config file that may point to the installation path: '/etc/.amAgentLocator'. Delete or edit this to match your new configuration to allow the setup script to operate properly

3. Before you begin the agentadmin setup, you'll need to create a file that contains the Agent password you entered when you set up OpenAM:

```
> cd /opt/web_agents
> echo "YourAgentPassword" > agentpass.txt
```

   a. Now when the agentadmin setup asks you for the file containing the password, you'd enter: /opt/web_agents/agentpass.txt
4. Depending on your Apache setup, you may not have an httpd.conf config file. This will make the agentadmin script tell you you didn't enter a proper Apache config path, so if it doesn't exist, go ahead and create it:

```
> touch /etc/apache2/httpd.conf
```

5. You will now run the agentadmin setup script, which will ask you questions regarding your instance so it can set up the agent

```
# First, stop the apache service if it's running
> service apache2 stop
* Stopping web server apache2
*
> bin/agentadmin --install

Please read the following License Agreement carefully:


[Press <Enter> to continue...] or [Enter n To Finish]

# Press 'n' to finish, and it will ask you if you agree:

Do you completely agree with all the terms and conditions of this License
Agreement (yes/no): [no]: yes # Type yes

...

******************************************************************************
Welcome to the OpenAM Policy Agent for Apache Server.
```

```
***************************************************************************

Enter the complete path to the directory which is used by Apache Server to
store its configuration Files. This directory uniquely identifies the
Apache Server instance that is secured by this Agent.
[ ? : Help, ! : Exit ]
Enter the Apache Server Config Directory Path [/opt/apache24/conf]: /etc/apache2
# NOTE this is the path to enter on Ubuntu, it varies on other Linux
distributions


Enter the URL where the OpenAM server is running. Please include the
deployment URI also as shown below:
(http://openam.sample.com:58080/openam)
[ ? : Help, < : Back, ! : Exit ]
OpenAM server URL: https://identity.yourserver.com:443/openam # Type in the
server, including protocol at the end of the domain, where your Identity server
is located, ending with the
                                                              # deployment
endpoint, usually /openam

Enter the Agent URL as shown below: (http://agent1.sample.com:1234)
[ ? : Help, < : Back, ! : Exit ]
Agent URL: https://data.yourserver.com:443 # Enter the base URL for your EM-API
installation on the Data VM


Enter the Agent profile name
[ ? : Help, < : Back, ! : Exit ]
Enter the Agent Profile name: DataVMAgent # Remember this name, as you'll need it
to set up Policies/Agents in OpenAM's GUI


Enter the path to a file that contains the password to be used for identifying
the Agent.
[ ? : Help, < : Back, ! : Exit ]
Enter the path to the password file: /opt/web_agents/agentpass.txt


WARNING:
Password validation cannot be done as OpenAM server is not running.

---------------------------------------------
SUMMARY OF YOUR RESPONSES
---------------------------------------------
Apache Server Config Directory : /etc/apache2
OpenAM server URL :
https://identity.yourserver.com:443/openam
Agent URL : https://data.yourserver.com:443
Agent Profile name : DataVMAgent
Agent Profile Password file name : /opt/web_agents/agentpass.txt
Verify your settings above and decide from the choices below.
1. Continue with Installation
2. Back to the last interaction
3. Start Over
4. Exit
Please make your selection [1]: 1

Creating directory layout and configuring Agent file for Agent_001
```

```
instance ...DONE.
Reading data from file /opt/web_agents/pass.txt and
encrypting it ...DONE.
Generating audit log file name ...DONE.
Creating tag swapped OpenSSOAgentBootstrap.properties file for instance
Agent_001 ...DONE.
Creating a backup for file /etc/apache2/httpd.conf ...DONE.
Adding Agent parameters to
/opt/home/nics/web_agents/apache24_agent/Agent_001/config/dsame.conf
file ...DONE.
Adding Agent parameters to /etc/apache2/httpd.conf file ...DONE.

SUMMARY OF AGENT INSTALLATION
-----------------------------
Agent instance name: Agent_001
Agent Bootstrap file location:
/opt/web_agents/apache24_agent/Agent_001/config/OpenSSOAgentBootstrap.properties
Agent Configuration Tag file location
/opt/web_agents/apache24_agent/Agent_001/config/OpenSSOAgentConfiguration.propert
ies
Agent Audit directory location:
/opt/web_agents/apache24_agent/Agent_001/logs/audit
Agent Debug directory location:
/opt/web_agents/apache24_agent/Agent_001/logs/debug

Install log file location:
/opt/web_agents/apache24_agent/installer-logs/audit/install.log
```

```
Thank you for using OpenAM Policy Agent

>
```

6. You should have seen the above successful installation with a Summary of files to take note of.
7. If you haven't already, start the apache service to verify the agent is installed

```
> service apache2 start
 * Starting web server apache2 # Ensure it started without error
 *
```

8. Depending on your Apache configuration, if you simply created httpd.conf without having any other configs refer to it, you'll need to copy the line the setup script added over to your apache2.conf file

```
> vi /etc/apache2/httpd.conf
# Look for a line like this:
include /opt/web_agents/apache24_agent/Agent_001/config/dsame.conf
```

   a. If you have that "include /..../dsame.conf line, copy that, and place it at the bottom of your apache2.conf file. Once you've done that, reload apache.
9. Now that apache2 is running again with the Agent properly being loaded, you should be able to open a browser, enter the data VM URL, and you should see a 403 Forbidden or be redirected to an OpenAM login screen.
10. You can now set up the Agent in the OpenAM Administration GUI.

> Be sure to return to OpenAM Configuration and to add the Custom header value to the API application, now that we've added the Agent.

Add Agent to OpenAM

Sign into the OpenAM Administrative GUI to add the Agent.

1. Navigate to Access Control-> / (Top Level Realm) -> Agents, and You'll be on the Web tab
2. Under the Agent table, click "New...". You'll be at a form called "New Web", where you'll enter the details for the Web Agent:
   a. Name: Enter the Profile Name you entered when you set up the Agent on the Data VM
   b. Password: Enter the password you set for the Agent when you initially set up the OpenAM Console
   c. Configuration: Leave as default, Centralized
   d. Server URL: Enter the full path, including port and deployment path, to your OpenAM instance, e.g., https://identity.yourserver.com:443/openam
   e. Agent URL: Enter the full path, including port, to your Data VM, e.g., https://data.yourserver.com:443 (the same as the Agent URL in the Agent setup steps)
   f. Click "Create"
   g. You're now back at the Web tab, with your Agent now listed in the Agents table.

Configure Policies

1. Navigate to Access Control-> / (Top Level Realm) -> Policies
2. Click "Add New Application"
3. Provide a Name and description: API
4. Click Next.
5. Define Resource Patterns
   a. There's a list of available patterns. By default, there's usually just one, "*". Click on that to add it to the column on the right.
   b. It'll highlight the asterisk *, and this is where you'll enter the base URL of the API: https://data.yourserver.com:443/* Note the asterisk on the end. Click the + button. It will add it to the Resources column.
   c. For the next one, which it has again highlighted the asterisk, enter the the full URL, including protocol, to your API deployment

path, e.g., https://data.yourserver.com:443/api/*, and click the + button.
    d. For this final one, enter the same URL as in the previous step, but also append "?*", so it looks like this: https://data.yourserver.com:443/api/*?*
    e. Click the + button.
    f. Click Next, then click Finish. You should be back at the Applications listing, now with your API entry added.
6. Find your "API" entry in the applications table. Note there's a pencil icon, and a page icon. If you hover over them with your mouse, the pencil says Edit Application, and the page says View Policies. Click on the Page/View Policies icon.
7. Click Add New Policy. You will be brought to a Policy creation form, where you can provide details for the policy.
    a. Name: API Policy
    b. Description: Policy governing API
    c. Click Next.
8. You should see a list of available Patterns, which you entered in the previous steps. Add each of them to the column on the right by clicking on them, then also clicking on the + button on the right to add it to the Resources. Click Next.
9. Check the Actions checkbox, which will select all actions for you, leave the default selection of Allow for each action, and click Next.
10. Click "+ Subject Condition", and in the Type dropdown that is added, select "Authenticated Users". Now you need to click and drag the box with this condition up into the green conditional box above, then click Next.
11. Click "+ Environment Condition", and in the Type dropdown, select "Authentication to a realm". A box will be provided. Enter "/" (without quotes), which is the default top level realm. Click and drag this box up into the green conditional box above, as in the previous step. Click Next.
12. We aren't setting any Repsponse Attributes, so click Next again.
13. Review your selections, and click Finish. You'll now see your Policy listed, with the resources it protects. Click Back.
14. Navigate to your Agent tab
    a. Click on your Agent in the Agent table, and click the OpenAM Services tab
    b. Scroll to the bottom of the page, in the Policy Client Service section. The last field is "Application". Enter the name of the Application you entered in the Policy for the API. In the above example, it was "API".
    c. Click Save.
15. Now to ensure everything has taken effect, restart the apache2 process on the Data VM. Now if you try to access your API via a browser, you should get redirected to an OpenAM login page.

Application Configuration

This guide describes how to get configuration files that work with your OpenAM instance. Specifically, for NICS6 Web, and EM-API.

Table of Contents

- Generate AMConfig.properties

Prerequisites

- Server Setup
- OpenAM Installation
- OpenAM Configuration
- Agent Installation

Generate AMConfig.properties

Get copy of AMConfig.properties generated by the Client SDK setup process

1. ssh into your Identity server where you've previously set up OpenAM, and where you have downloaded the OpenAM packages to.
2. This guide will assume you have your ClientSDK-12.0.0.zip in /opt/openam on your Identity VM.
3. Go to your openam directory, and unzip the Client SDK files:

```
> cd /opt/openam
> unzip ExampleClientSDK-CLI-12.0.0.zip -d clientSdk # Be sure to specify a
directory, otherwise it will just extract into your current directory
> cd clientSdk
```

4. Now that you're in your newly created "clientSdk" folder, we're going to run the setup script:

```
> scripts/setup.sh
Debug directory (make sure this directory exists): /opt/openam/clientSdk/debug
Application user (e.g. URLAccessAgent) password: (enter your Agent password)
Protocol of the server: https
Host name of the server: identity.yourdomain.com
Port of the server: 443
Server's deployment URI: openam
Naming URL (hit enter to accept default value,
https://identity.yourserver.com:443/openam/namingservice): <Enter> # Be sure the
default it generates is accurate
>
```

5. You'll now have some extra files in your clientSdk directory. In particular, we want the AMConfig.properties file in the resources directory. Skim this file to see that it has references to your Identity server's hostname, etc. This is the file you want to copy to /opt/data/nics/config on the Web and Data VMs.

OpenAM Configuration

Guide for configuring OpenAM for use with NICS6.

Prerequisite(s)

- Server Setup
- OpenAM Installation

Contents

- Configuration Changes
  - Add the 'nics.admin' group:
  - Give the nics.admin group elevated privileges:
  - Create the "nicsadmin" user, and make them part of the "nics.admin" group
  - Remove the 'anonymous' user, and change the demo user's password
  - Adjust session timeouts
  - Add the CUSTOM-uid header

Configuration Changes

Add the 'nics.admin' group:

1. From the OpenAM console, click the Access Control tab. Then click the realm name you wish to configure. By default it's "/ (Top Level Realm)"
2. Click the Subjects tab, then click on the Group tab.
3. Click "New...", and enter "nics.admin" as the ID. Click "OK". You should now have a "nics.admin" group in the Group table.

Give the nics.admin group elevated privileges:

1. From the Access Control-> / (Top Level Realm) screen, click on the Privileges tab, and click on "nics.admin" in the Privileges table.
2. Check the following privileges
   a. Read and write access to all realm and policy properties
   b. REST calls for reading realms
   c. REST calls for reading subject conditions
   d. REST calls for reading subject attributes
3. Click Save. Your "nics.admin" group should now have the above privileges. You should see a message saying the Privilege Profile was updated if it worked.
4. Click Back to Privilege(s)

Create the "nicsadmin" user, and make them part of the "nics.admin" group

1. Navigate to Access Control-> / (Top Level Realm), ensure you're on the User tab, and click New....
2. Fill out the following fields: See the tip below regarding encrypting an admin user for use in configuration files.
    a. **ID:** nicsadmin
    b. **First Name:** NICS
    c. **Last Name:** Admin
    d. **Full Name:** NICS Admin
    e. **Password:** (Choose a password)
    f. **Password (confirm):** (Choose a password)
    g. **User Status:** Active
3. NOTE: The password for this user will have to undergo an encryption routine so that you can specify this user and password in configuration files. Thus far it has been left as the default testing password, since the encryption tools are outdated. NEED TO FIX.
4. Click OK. You should now see the "NICS Admin" user in the User table.
5. Click on the NICS Admin user. Click on the Group tab. Select the "nics.admin(nics.admin)" entry in the Available column, and click the "Add >" button to apply it to the Selected column. Click Save, and you should see a message saying the Profile was updated.
6. Click Back to Subjects.

> The nicsadmin user is used by applications that interact with OpenAM, like IWEB and EMAPI. Their configuration files contain encrypted properties for the admin user and the admin password. We have an older build of the user-management-tools application that can encrypt and decrypt strings in the manner OpenAM needs to use them. See OpenAM User/Pass Encryption for details.

Remove the 'anonymous' user, and change the demo user's password

1. Navigate to Access Control -> / (Top Level Realm), and click the User tab
2. Check the checkbox next to the 'anonymous' user, and click the Delete button
3. Click on the 'demo' user. Click the 'Edit' link next to Password, which opens up a change password dialog with the original password box greyed out.
4. Enter a new password and click OK, and close the dialog.

Adjust session timeouts

1. From the home screen (Common Tasks), click the Configuration tab, then click the Global tab, then the "Session" service in the Global Properties table.
2. Scroll down to the bottom of the Session configuration screen to the Dynamic Attributes section. The values entered here depend on your preferences. For NICS we currently use the following values:
    a. Maximum Session Time: 720
    b. Maximum Idle Time: 719
    c. Maximum Caching Time: 3
    d. Active User Sessions: 100
3. Click Save

Add the CUSTOM-uid header

NOTE: Configure after setting up the agent

1. Navigate to Access Control -> / (Top Level Realm), and click the Agents tab
2. Click on the Agent you configured for the API
3. Click on the Application tab
4. Scroll down to the Profile Attributes section
    a. Profile Attribute Fetch Mode : HTTP_HEADER
    b. New Value : uid
    c. Custom Value : CUSTOM-uid
    d. Click Add
5. Click Save

> The username associated with the token making the request is passed to the API in the header. This provides further security as the user is validated to ensure he/she has permissions to request or post information to the API.

OpenAM User/Pass Encryption

The most simple way is to use the "ampassword" script provided by OpenAM in the SSOAdminTools package.

> The current OpenAM version NICS6 uses, as of 2016-02-03, is **OpenAM 12.0.0**

Once you've acquired the SSOAdminTools-<VERSION>.zip, perform the following steps:

1. Unzip with your favorite zip management software to new directory: "ssoAdminTools"
2. CD into the directory, and run the "setup" script

```
> ./setup
...
Do you accept the license? yes
Path to config files of OpenAM server [/root/openam]: /usr/share/tomcat8/openam
Debug Directory [/opt/home/nics/openam/ssoAdminTools/debug]: <Enter to accept
default>
Log Directory [/opt/home/nics/openam/ssoAdminTools/log]: <Enter to accept
default>
The scripts are properly setup under directory: /opt/openam/ssoAdminTools
Debug directory is /opt/openam/ssoAdminTools/debug.
Log directory is /opt/openam/ssoAdminTools/log.
The version of this tools.zip is: OpenAM 12.0.0
The version of your server instance is: OpenAM 12.0.0 Build 11961
(2014-December-17 21:16)
>
```

3. Now in your ssoAdminTools directory will be an "openam/bin" directory with scripts in it, including an "ampassword" script
4. Place the string you want to encrypt into a file. This can be the username or the password you wish to encrypt
5. Run the following commands:

```
> openam/bin/ampassword -e <path to file containing password>
```

6. The script will output the encrypted value for use in your OpenAM configuration files

Example

The following assumes that the full OpenAM download was extracted to /opt/openam

```
> cd /opt/openam/ssoAdminTools
> echo "MyPassword" > mypass.txt
> ./openam/bin/ampassword -e mypass.txt
AQICTv0GCVx/JcAaLd7cmxSUyJa3e8jFtleU
```

OpenAM Installation

Guide for deploying OpenAM, and initial configuration.

Prerequisite(s):

- Server Setup
- Acquire OpenAM Packages

Step-by-step guide

Initial OpenAM installation

1. After having acquired the war (see Acquire OpenAM Packages), copy it to Tomcat

```
> sudo cp OpenAM-12.0.0.war /var/lib/tomcat8/webapps/openam.war
```

    a. Note that we copied to "openam.war" so that it'll show up in your URL as /openam
2. If you've properly set up Apache, you should now be able to access the deployed OpenAM instance by navigating to the URL in a browser, e.g., https://identity.yourhost.com/openam. You should see the OpenAM configuration page asking if you want the default or the custom configuration. Click the link under Custom Configuration, "Create New Configuration".
3. Read and accept the license agreement by checking "I accept the license agreement" and clicking Continue.
4. Enter the password you would like to use for the 'amadmin' user
5. Server Settings
    a. Ensure the Server URL matches your FQDN (which you should have set up prior to deploying the openam.war). Also change the protocol and port from http:80 to https:443 if you're using SSL (Recommended).
    b. Cookie Domain should be auto-filled with ".yourdomain.tld", e.g., ".yourdomain.com"
    c. Configuration directory can be left as the default: /usr/share/tomcat8/openam, which is tomcat's home directory on Ubuntu running Tomcat8.
    d. Click Next
6. Configuration Data Store Settings
    a. Leave the default "First Instance" radio button selected
    b. Leave the "OpenAM" radio button selected for Configuration Data Store (again, unless you're using some other external instance)
    c. Copy/Write down the encryption key in case you want to use it for accessing settings with OpenAM admin tools (can be used to encrypt/decrypt configuration exports, etc)
    d. Usually you'd alter the root suffix to match your FQDN, but since this is just an internal configuration store, you can leave it as the default. If you were creating a new external store, or making use of a pre-existing one, you'd need to enter the information for that datastore instead.
    e. Click Next.
7. User Data Store Settings
    a. It's not recommended to use the embedded OpenDJ user store in production environments, but so far that's what NICS has been using. Once a fully set up external LDAP/OpenDJ store is tested, we will update documenation.
    b. Select the "OpenAM User Data Store" radio button
        i. You'll see the warning: **The OpenAM user data store is not recommended for large scale production environments or deployments with a complex topology.**
    c. Click Next.
8. Site Configuration
    a. Currently NICS does not set up load balancing within OpenAM. Leave the default of "No" selected, and click Next.
9. Default Policy Agent User
    a. Choose and enter a password for the Policy Agent. This will be used when setting up the Web Agent for Policy enforcement later.
    b. Click Next.
10. Review your configuration choices, and if satisfied, click Create Configuration.
11. If configuration completed successfully, you should see a dialog saying "Configuration Complete". Click the link to "Proceed to login".
12. You should be brought to the OpenAM login page. If you weren't, you probably didn't set up your FQDN properly before configuring OpenAM.
13. Login with the 'amadmin' user, which you specified the password for in step 4.
14. You should now be logged into the OpenAM Administration Console.

Server Setup

Pre-Install

JVM Tuning

As the guide recommends, look at their JVM Tuning document, and set up Tomcat 8 accordingly: https://backstage.forgerock.com/#!/docs/openam/12.0.0/admin-guide/chap-tuning

To set the JVM options:

1. Check if your Tomcat8 install already has a setenv.sh script in /usr/share/tomcat8/bin
    a. If it doesn't, go ahead and create the setenv.sh file, be sure to change the owner to tomcat, and give it execution permission:

Tomcat setenv.sh Creation

```
> cd /usr/share/tomcat8/bin
> touch setenv.sh

# NOTE: the user:group you specify here should match the other files in the
Tomcat directory. Depending on your installation, you may
#        have a different tomcat group

> chown tomcat8:TOMCAT setenv.sh
> chmod u+x setenv.sh
```

    b. If it already does, ensure it has execution permission and is owned by tomcat, and continue to the next step.
2. Using your favorite command line editor, open setenv.sh and edit it to match the following. If you already have a setenv.sh script, then use your discretion for options not specified below that you may have previously set.

setenv.sh contents

```
JAVA_OPTS="-server -Xms2048m -Xmx2048m -XX:NewSize=256m -XX:MaxNewSize=256m
-XX:PermSize=256m -XX:MaxPermSize=256m -Dsun.net.client.defaultReadTimeout=60000
-XX:+UseConcMarkSweepGC -XX:+UseCMSCompactAtFullCollection
-XX:+CMSClassUnloadingEnabled"
```

    a. Note that we're using the recommended 2048m due to the use of OpenDJ. IF you're NOT using OpenDJ, then 1024m should be sufficient. Although they do recommend even higher for a production environment.
3. Once you've saved the setenv.sh file to disk, restart Tomcat, and verify it's using your specified settings

```
> service tomcat8 stop # if it was already running
 * Stopping Tomcat servlet engine tomcat8
[ OK ]
> service tomcat8 start
 * Starting Tomcat servlet engine tomcat8
[ OK ]
> psaux | grep tomcat
tomcat8  18364  0.1 29.1 3842824 1179324 ?     Sl   Jan29   4:18
/usr/lib/jvm/default-java/bin/java
-Djava.util.logging.config.file=/var/lib/tomcat8/conf/logging.properties
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -server
-Xms2048m -Xmx2048m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:PermSize=256m
-XX:MaxPermSize=256m -Djava.endorsed.dirs=/usr/share/tomcat8/endorsed -classpath
/usr/share/tomcat8/bin/bootstrap.jar:/usr/share/tomcat8/bin/tomcat-juli.jar
-Dcatalina.base=/var/lib/tomcat8 -Dcatalina.home=/usr/share/tomcat8
-Djava.io.tmpdir=/tmp/tomcat8-tomcat8-tmp org.apache.catalina.startup.Bootstrap
start
```

    a. Note we can see it's using our JVM parameters

b. If Tomcat failed to start, check your JVM parameters. Also check your server's actual memory capacity. Tomcat won't start if the Xms or Xmx are set higher than the actual available memory.

4. Next, make sure that desired JDK is being used. In this case we can see that /usr/lib/jvm/default-java/bin/java is being used. Make sure this link is pointing to a JDK 1.7 Java installation:

```
> ls -l /usr/lib/jvm/default-java
lrwxrwxrwx 1 root root 11 Jun  2  2015 /usr/lib/jvm/default-java -> jdk1.7.0_79/
```

a. If it's not pointing to an Oracle JDK installation, you should update the default-java link to point to one if you have one, and install one if you don't.

File Descriptors

Since we plan to use OpenDJ, the guide also recommends we increase the file descriptors available to the openam user (TODO: Shouldn't this happen for the tomcat user?)

1. Edit /etc/security/limits.conf and add the following lines (if they're not already there):
/etc/security/limits.conf

```
openam soft nofile 65536
openam hard nofile 131072
```

Apache

Configure Apache to forward to /openam

1. Add the following to your HTTP site config, e.g., /etc/apache2/sites-available/default-ssl.conf

```
Redirect permanent /openam https://[public FQDN]/openam
 Redirect permanent / https://[public FQDN]/openam
```

a. Note that this assumes you'll never want to access the site over http, just https, which is the recommended setup.
b. [public FQDN] should be your full public URL to the server that matches the SSL certificates, e.g., identity.nics.ll.mit.edu
2. Add the following to your SSL site config, e.g., /etc/apache2/sites-available/default-ssl.conf:

```
ProxyPass /openam http://[host/IP]:8080/openam
ProxyPassReverse /openam http://[host/IP]:8080/openam
ProxyPass / http://[host/IP]:8080/openam
ProxyPassReverse / http://[host/IP]:8080/openam
```

a. Where [host/IP] is the internally accessible hostname or IP address of the server

If it's not already enabled, you'll need to enable the Proxy module for Apache to use the ProxyPass and ProxyPassReverse lines

Setting your FQDN (Fully Qualified Domain Name)

1. First, check to see if your FQDN is already set correctly:

```
> hostname -f
identity.yourhost.com # Example result
```

2. If your result shows the FQDN, such as identity.yourhost.com, the FULL URL used to access your server publicly, then your FQDN is already set properly for OpenAM. If not, follow the steps below.
3. If your FQDN is not set correctly, with sudo privileges, edit your /etc/hosts file. There's usually a line like "127.0.1.1 localhost". Edit that to the following:

```
<the IP of this server> <your FQDN> <the hostname of the server>
```

4. Example:

```
123.45.6.7    identity.somehost.com    nics-identity
```

    a. Now when you run "hostname -f", with the example above, you would see "identity.somehost.com"
5. This line doesn't have to remain once you've configured OpenAM, but it's required during initial setup of OpenAM to properly populate fields and configuration files.
6. Further reading: Getting Started - Prepare etc hosts