Install & Configure the Web VM

The NICS web application is deployed to tomcat on a server that is also running RabbitMQ. The RESTful API (em-api) can be deployed on the same server or a different tomcat server. The instructions are the same but the em-api config files need updated to point at the correct RabbitMQ message bus. See an explanation of configuration properties in the NICS Configuration Properties section below.

Prerequisites

- Apache 2.4.7
- Tomcat 8
- Rabbitmq 3.4.4
- Oracle JDK 7

Load the proxy_wstunnel module

- sudo a2enmod proxy_wstunnel

Configuration Files

1. **/nics-web/web-app/src/main/config/core.properties**

The core.properties is located in /opt/data/nics/config

```
endpoint.rest=https://<hostname of data vm>/api/v1
endpoint.geoserver=https://<hostname of geoserver vm>/geoserver
#Example .nics.com
private.cookie.domain=<cookie domain>

iplanet.key=iPlanetDirectoryPro
iplanet.path=/
iplanet.domain=<same as the cookie domain>

openam.key=AMAuthCookie
openam.path=/
openam.domain=<same as the cookie domain>

rabbitmq.hostname=localhost
rabbitmq.username=guest
rabbitmq.userpwd=guest
rabbitmq.exchange.name=iweb.amq.topic

# In minutes
token.timeout=720

# Report properties. Storage isn't really used on the web side, so will factor out at some point
report.general.storage=/opt/data/nics/static/reports/general/
report.general.url=https://<web vm>/static/reports/general/

report.damage.storage=/opt/data/nics/static/reports/damage
report.damage.url=https://<web vm>/static/reports/damage/
```

2. **apache config**

```
Alias /static /opt/data/nics/static
ProxyPass /static !


ProxyPass /nics/mediator ws://<web ip>:8080/nics/mediator
ProxyPass /nics http://<web ip>:8080/nics
ProxyPassReverse /nics http://<web ip>:8080/nics

ProxyPass /em-api https://<data vm>/api
ProxyPassReverse /em-api https://<data vm>/api

ProxyPass / http://<web ip>:8080/nics
ProxyPassReverse / http://<web ip>:8080/nics

ProxyPass /static !
<Directory "/opt/data/nics/static">
        Require all granted
</Directory>
```

Configure the NICS static data directory and root redirect

We have a data directory on the Data VM for serving up static file content. Uploaded datalayers and other documentation are stored in this directory. We need to configure Apache to serve up this directory and mount the directory locally. The directory should be `/opt/data/nics/static`.

1. Edit the file `/etc/apache2/httpd.conf` and add the following lines:

```
# Web-enable the static directory
Alias /static /opt/data/nics/static
```

2. Finally restart the apache2 service to make sure our settings changes take effect:

```
> service apache2 restart
 * Restarting web server apache2
 ... waiting                                                          [ OK ]
```

If there are problems with the settings the restart will fail.

Mount the data vm to web vm

- Edit the file /etc/fstab as root
- At the end of the file add the line:

  *<ip of vm mounting from>***:***<path to folder to mount from>   <local path to folder to mount to>*   **nfs rw,user,auto,exec 0 0**

  Example:

  <data vm ip>:/opt/data/nics/upload /opt/data/nics/static   nfs   rw,user,auto,exec 0 0

  NOTE: the folder you are mounting to locally must already be created

  NOTE2: there must be a newline at the end of the fstab file

- Mount the folder (NOTE: This step assumes you've already configured the mount on the Data VM)

  sudo mount -a

RabbitMQ Loopback Config

- Starting with RabbitMQ version 3.3.0, the default 'guest' user is no longer allowed to connect to a RabbitMQ instance remotely. The workaround is to either create a new RabbitMQ user for your connections, which is recommended over the 'guest' user anyway, or add a config to allow all users to connect remotely.
- Add the following to your /etc/rabbitmq/rabbitmq.config file. If the file does not exist, create it:

  [{rabbit, [{loopback_users, []}]}].
- Note: The period at the end is required.
- Once you've added this to rabbitmq.config, you'll have to restart your rabbitmq server for it to pick up the changes. Once you restart, you should now be able to connect to this instance remotely with the 'guest' user.

Deploy NICS

- deploy nics.war to /var/lib/tomcat8/webapps

Configure and Install CollabFeedManager:

The collabfeed manager runs on the web machine and listens for feature updates to collaboration rooms. It creates a datalayer in geoserver that can be imported into and exported from nics.

1. Run the following commands in the / directory. Create a directory to run nics components from.

   ```
   > mkdir -p /opt/nics/deploy
   ```
2. Untar the file

   ```
   # Untar all files to nics deploy
   > tar -xvzf collab-feed-manager-X.X.X.tar.gz .
   ```
3. Configure properties file
   a. Open the properties file:

      ```
      > vi /opt/nics/deploy/collab-feed-manager/collabFeedManager.properties
      ```
   b. Configure: dbHost should point to a database vm instance used by the MapServer

      ```
      dbName=nics
      dbUsername=postgres
      dbPassword=postgrespassword
      dbHost=<host ip>
      dbPort=<host port>
      geoserverUrl=http://<geoserver ip>:8080/geoserver/rest
      geoserverUsername=admin
      geoserverPassword=geoserverpassword
      workspaceName=<workspace name> #this needs to be created in geoserver - refer to the g
      dataStoreName=<store name> #this need to be created in geoserver
      syncInterval=3600
      kmlPublishInterval=10
      kmlTimeout=120
      collabSrcUrl=rabbitmq://localhost:5672?amqExchange=amq.topic&amqExchangeType=topic&re
      lse&msgContentType=text
      ```
4. Start the component

```
# Copy collab-feed-manager to nics deploy
> cd /opt/nics/deploy/collab-feed-manager/
> nohup ./start.sh > logs/collab-feed-manager.log &
```