Multiple Web Front Ends to a Single XNAT Database or Archive (Multi-Node XNAT)

It is possible to have multiple XNAT instances pointing to the same XNAT database and archive. This setup of multiple web front ends can be used to offload handling of some requests from the primary XNAT web instance in order to maintain performance of the primary web application. These secondary servers commonly are referred to as *shadow servers* and are often used to handle application server intensive requests such as the receiving of DICOM, building of sessions from the prearchive and the handling of REST-based communications with pipelines or other batch processing jobs. Typically, these shadow servers are running the identical version of the XNAT web application as the primary server and are often placed behind the organization's firewall, since they're expected to handle internal processing and receiving of sessions rather than usage by standard users. This is not a load balanced set up as described in this article, where requests to the same URL are passed among a number of identical back end nodes. Rather, in this case, each server is reached by a different URL, so the determination of which server to use is determined by the pipeline, task process, or configured in the scanner or CTP. The setup of shadow servers is simple, however there are issues that must be considered and taken into account by the configuration. A new feature in XNAT 1.7, the node/task framework, is introduced in XNAT 1.7 to help resolve some of these issues.

Issues

For the most part, having multiple web front-ends in your installation should not pose issues. In most cases, users make requests to a specific node, and that node responds and handles the request. Some processes, however, are not triggered by a user request, but are background processes waiting to respond to something to that happens on the system or are scheduled processes running at specified intervals. It's often important that multiple nodes aren't trying to respond to the same event, and we may want to control which node, in a multi-node environment, responds.

In an out-of-the-box XNAT installation, one such process is the **Session XML Rebuilder**. The Session XML Rebuilder monitors the prearchive space for incoming DICOM and builds XNAT sessions from that DICOM. **It is important that multiple nodes not try to build the same session at the same time** .

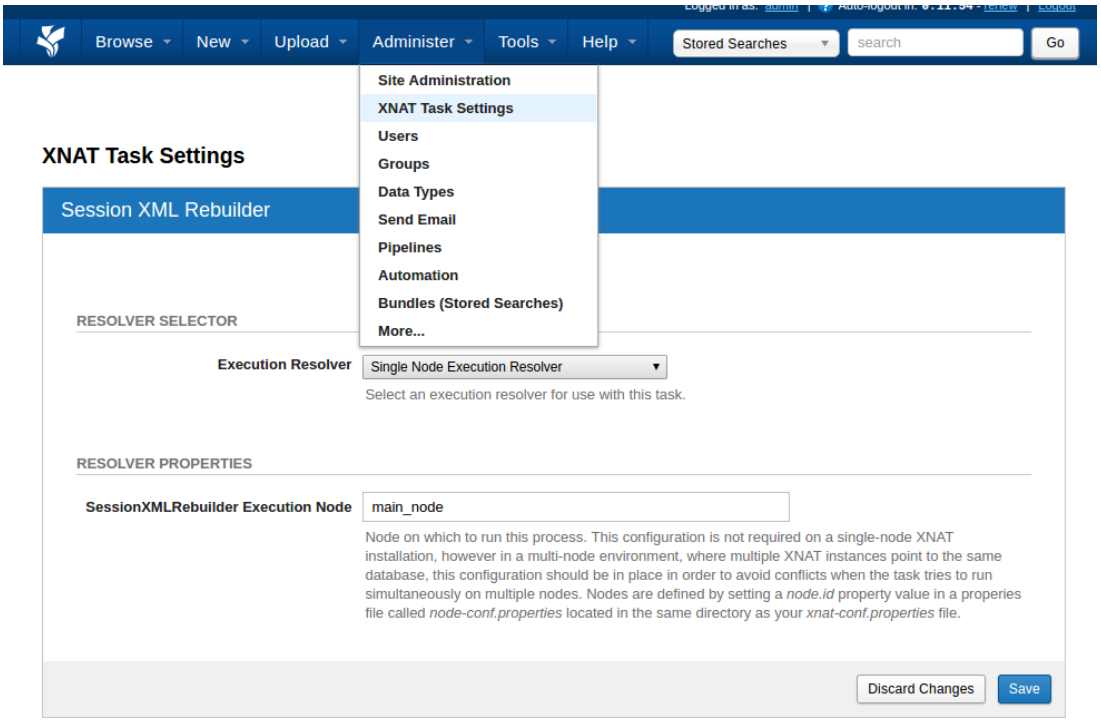
Setup

For the most part, setup of a multi-node XNAT differs little from setup of a single-node instance. In a multi-node instance, there's just an additional configuration file and an additional configuration page to complete a proper set up. To configure each XNAT instance, you'll follow the steps found in the getting started guide. Rather than having the **xnat-conf.properties** file point to different databases, however, in a multi-node environment all nodes will point to the same database. Each node, then, will need to have the configured **File System** locations (which will be the same, since they'll be pulling them from the same database) mounted at the correct location. Once your nodes are set up to point to the same database and file system locations, there is just a little more setup and configuration to complete your installation.

To indicate that your instances are part of a multi-node installation of XNAT, you'll add a file called **node-conf.properties** in the config directory of each instance. The config directory is in the directory where your **xnat-conf.properties** file is located on each instance. The **node-conf.properties** file simply contains the specification of a **node.id**, distinct for each node, as follows:

node.id=node-name

That's all that file needs to contain. If presence of a **node-conf.properties** file after a Tomcat restart indicates you are operating in a multi-node environment, site-admin users will see an *XNAT Task Settings* item under the *Administer* tab. This item and the resulting page is shown in the screenshot below:



The Node/Task Framework uses *execution resolvers*, which, based on configuration, decide whether a given node should complete a task. You can choose from a couple of execution resolvers in an out-of-the-box XNAT installation. One resolver (shown) allows specification of a single node that will run the task. The resolver not shown allows specification of a list of nodes to allow for fail-over.

Currently, in an out-of-the box installation of XNAT 1.7.X, the session XML rebuilder is the only task identified as requiring special handling in a multi-node environment. In a customized environment, there might be configuration for additional tasks that have been defined in plugins.

Additional Notes

Earlier versions of XNAT could have issues with accession numbers, in a multi-node setup, when new experiments were generated on different nodes at nearly the same time. Multiple experiments could receive the same accession number. This issue has been resolved in XNAT 1.7 by keeping track of the nodes and modifying the accession number slightly for sessions created on other nodes.

Whereas sessions created on the first node, might be something similar to the following:

XNAT_E23456

Additional nodes will have accession numbers similar to the following:

XNAT02_E00001, XNAT03_E00002